ANALISIS *ROUTING MULTI COPY* DENGAN *STATIONARY RELAY NODE* DAN *MANAGEMENT BUFFER FIRST IN – FIRST OUT (FIFO)* PADA *DELAY TOLERANT NETWORK (DTN)*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Muhammad Reza Wahyu Chrisdyan
NIM: 135150207111082

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

Analisis *Routing Multi Copy* Dengan *Stationary Relay Node* dan *Management Buffer First In – First Out* (FIFO) Pada *Delay Tolerant Network* (DTN)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muhammad Reza Wahyu Chrisdyan
NIM: 135150207111082

Skripsi ini telah diuji dan dinyatakan lulus pada
3 agustus 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Rakhmadhany Primananda, S.T., M.Kom	**Reza Andria Siregar, S.T., M.Kom**
NIK: 201609 86040 6 1001	**NIP: 19790621 200604 1 003**

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D
NIP: 19710518 200312 1 001

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, didalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik disuatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 03 Agustus 2018

Muhammad Reza Wahyu Chrisdyan

NIM: 135150207111082

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

KATA PENGANTAR

Dengan menyebut nama Allah SWT yang maha pengasih dan maha penyayang, penulis ucapkan puji syukur atas kehadirat Allah SWT karena dengan rahmat dan hidayah-Nya, penulis dapat menyelesaikan skripsi yang berjudul **"ANALISIS *ROUTING MULTI COPY* DENGAN *STATIONARY RELAY NODE* DAN *MANAGEMENT BUFFER FIRST IN – FIRST OUT (FIFO)* PADA *DELAY TOLERANT NETWORK (DTN)*".** Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Fakultas Ilmu Komputer Universitas Brawijaya Malang.

Penulis menyadari bahwa, skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak yang memberi semangat, kritik, saran, bimbingan dan doa. Oleh karena itu ucapan terima kasih penulis sampaikan kepada:

1. Allah S.W.T atas berkah, rahmat dan hidayahnya yang senantiasa menyertai dalam pelaksanaan skripsi ini.
2. Bapak Rakhmadhany Primananda, S.T. M.Kom selaku dosen pembibing I yang telah sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
3. Bapak Reza Andria Siregar, S.T., M.Kom selaku dosen pembimbing II yang telah bersabar dengan membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. Selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Agus Wahyu Widodo, S.T,M.Sc selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
6. Ayah Iswahyudi dan Ibu Arista atas segala nasehat, kasih sayang, perhatian dan kesabarannya dalam mendidik penulis, serta yang senantiasa tiada henti-hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
7. Seluruh keluarga besar Anindya Cahya Faticha yang telah memberikan waktu, perhatian dan kesabarannya untuk menyemangati agar cepat wisuda, serta senantiasa tiada henti-hentinya memberikan doa.
8. Seluruh teman – teman Informatika angkatan 2013 Universitas Brawijaya yang telah banyak memberi masukan dan semangat kepada penulis sehingga terselesaikan skripsi ini.

Semoga jasa dan amal baik mendapatkan balasan dari Allah SWT. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya terutama mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya.

Malang, 03 Agustus 2018

Penulis

m.rezawahyu.c@gmail.com

repository.ub.ac.id

ABSTRAK

Pada pengiriman data digital yang dilakukan, *delay* yang besar merupakan salah satu faktor gagalnya penerimaan pesan. Pada permasalahan tersebut, salah satu teori yang bisa diterapkan adalah jaringan DTN (*Delay Tolerant Network*). DTN memiliki keunggulan saat diterapkan pada jaringan dengan karakteristik *delay* panjang, tingkat *loss* tinggi, dan tingkat konektivitas rendah serta infrastruktur yang minim. Jenis *routing Multi Copy* pada DTN dapat memaksimalkan rasio pengiriman dan mengurangi penundaan tetapi membutuhkan tambahan sumber daya jaringan. Banyak penelitian yang mengasumsikan sumber daya jaringan yang tak terbatas, salah satunya yaitu *buffer.* DTN dapat mengimplementasikan *Management Buffer Policy* sebagai pengoptimalan *buffer* yang terbatas saat pengiriman pesan. *Stationary Relay Node* pada DTN dapat digunakan untuk meningkatkan probabilitas penerimaan pesan serta jembatan untuk pengiriman pesan. Pada penelitian kali ini membandingan protokol DTN *routing Multi Copy* dengan menambahkan *Stationary Relay Node* dan *Management Buffer* dengan algoritma *First In - First Out* yang disimulasikan menggunakan *ONE Simulator.* Hasil pengujian dianalisis menggunakan parameter performansi *Delivery Probability* dan *Overhead Ratio* dengan skenario menggunakan *routing Epidemic, ProPHET* dan *Spray and Wait* dengan *node* yang bergerak sebanyak 150 dan *Stationary Relay Node* sebanyak 10, ukuran pesan sebesar 5Mb dan batasan *buffer* sebesar 30Mb, 50Mb, 80Mb dan 100Mb. Dari hasil pengujian dan analisis yang telah dilakukan dapat diambil kesimpulan bahwa *routing Spray and Wait* efektif untuk menyampaikan pesan tetapi dengan beban jaringan yang lebih rendah dari *routing* lain yang diujikan yaitu dengan nilai *Delivery Probability* terbesar 0,2464 dan *Overhead Ratio* sebesar 10,85%.

Kata kunci: *Delay Tolerant Network, routing protocol multi copy, Epidemic, ProPHET, Spray and Wait, management buffer, Algorithm First in – First Out, Stationary Relay Node.*

UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

ABSTRACT

On the delivery of digital data, a big delay is one of the factors in failure of the receipt of message. On this problems, one theory that can be applied is a DTN network (delay tolerant networking). DTN has advantages when its applied to network with long delay characteristics, the high degree of loss, and the low level of connectivity and minimal infrastructure. Multi Copy routing types on DTN can maximize the ratio of delivery and reduce delays but requires additional network resources. A lot of research that assumes a network resource is infinite, one of them is buffer. DTN can implement Buffer management policy as a limited buffer optimization when sending messages. Stationary Relay Node at DTN can be used this way to increase probability of the acceptance of the message as well as a bridge for sending messages. At this time, the research protocol comparing DTN routing Multi Copy by adding Stationary Relay Node and Buffer management algorithm with a first in-first out were simulated using a Simulator. The test results are analyzed using the performance parameters of the probability of shipping and Overhead ratios with scenarios using epidemic routing, prophets and spray and wait by moving as many as 150 nodes and Node Relays as many as 10 Stationary, the message size of 5 MB and 30 MB of buffer limitation, 50 MB, 80 MB and 100 Mb. From the results of testing and analysis can be concluded that routing spray and wait is effective to convey the message with the network load that is lower than the other routing that applied with the greatest probability value delivery 0.2464 and Overhead ratio amounted to 10.85%.

Keyword : *Delay Tolerant Network, routing protocol multi copy, Epidemic, ProPHET, Spray and Wait, management buffer, Algorithm First in – First Out, Stationary Relay Node.*

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

DAFTAR GAMBAR

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR TABEL

BAB 1 PENDAHULUAN

1.1 Latar belakang

Internet (*Interconection Networking*) merupakan jaringan komputer yang saling terhubung satu sama lain. Agar dapat mengirimkan data digital, pada jaringan internet tidak boleh terjadi atau mempunyai *delay* yang besar. Apabila hal tersebut terjadi maka data yang dikirimkan akan berhenti pada satu titik *node* dan selanjutnya data akan dibuang jika batas waktu pesan di jaringan melebihi waktu normal (Restu & Yovita, 2015). *Delay* dapat didefinisikan sebagai perhitungan atau selisih waktu pada saat pengiriman pesan (sebuah bit atau paket data) dari sumber pesan kepada penerima (Fajri, 2016). Pada pengiriman data digital yang dilakukan, *delay* yang besar merupakan salah satu faktor gagalnya penerimaan pesan. Pesan yang dikirim akan terhambat dan hilang karena *delay* yang besar dapat menghabiskan waktu aktif pesan dalam jaringan. Beberapa kasus pengiriman data di daerah terpencil yang hanya mempunyai infrastruktur teknologi informasi yang terbatas, data tidak dapat dikirimkan karena *delay* yang terlalu besar serta *node* yang tidak selalu bisa saling berkomunikasi.

Pada permasalahan tersebut, salah satu solusi yang bisa diterapkan adalah jaringan DTN (*Delay Tolerant Network*). Dengan menggunakan DTN, layanan internet pada daerah tertentu dapat diterapkan dan disajikan di daerah atau area yang memiliki karakteristik *delay* panjang, tingkat *loss* tinggi, dan tingkat konektivitas rendah serta infrastruktur yang minim (LAP, 2010). Pada DTN mekanisme jaringan yang digunakan dalam pengambilan data yaitu dengan *fixed data network, dynamic network* dan jaringan antar *node* yang terputus (Warthman, 2012). Melalui pengaturan parameter yang tepat, rasio waktu pengiriman dan tenggang waktu pengiriman, dapat dicapai sebagian waktu dengan metode yang diusulkan (LAP, 2010). DTN memiliki beberapa algoritma *routing* sebagai pertukaran data yang terjadi antar *node*, setiap *routing* mempunyai karakteriktik pertukaran data yang berbeda satu dengan yang lain sesuai dengan skema dan jenis algoritma *routing*nya.

Terdapat dua jenis algoritma *routing* yang dapat digunakan pada DTN, yaitu skema *routing Single Copy* dan skema *routing Multi Copy*. Skema *routing Single Copy* yaitu dengan mengirimkan pesan yang uniq dan diteruskan disepanjang jalur pengiriman. Pada jenis skema *routing Multi Copy* merupakan jenis protokol *routing* yang meneruskan setiap pesan pada *node* dibanyak jalur yang tersedia. Dengan demikian dibandingkan dengan jenis *routing Single Copy*, jenis *routing Multi Copy* dapat memaksimalkan rasio pengiriman dan mengurangi penundaan tetapi membutuhkan tambahan sumber daya jaringan yaitu *energy, bandwith, storage* dan *buffer* (Muis, 2013).

Pada penelitian sebelumnya yang dilakukan oleh (Poltak, 2017), dengan judul Perbandingan Kinerja *Routing Multi Copy* dan *Routing First Contact* Dengan *Stationary Relay Node* Pada *Delay Tolerant Network* (DTN). Penelitian tersebut menggunakan protokol Routing *Multi Copy* dan *First Contact* yang disimulasikan

menggunakan aplikasi *ONE Simulator* dan mengambil rute jalur perdakian ke gunung Semeru. Penelitian tersebut menganalisis *routing* DTN *routing Multi Copy* dan *First Contact* dengan melakukan penambahan *Stationary Relay Node* akantetapi pada penelitian tersebut mengansumsikan *buffer* yang tidak terbatas dalam melakukan pengiriman data. Sedangkan dibanyak aplikasi serta teknologi yang dimanfaatkan pada teknologi DTN memiliki keterbatasan dalam *buffer* dan energi. Sehingga pada penerapan yang sesungguhnya dipenelitian sebelumnya masih memiliki kelemahan dalam hal batasan sumberdaya seperti ukuran *Buffer.* Oleh karena itu pada penelitian kali ini melakukan perbandingan protokol DTN *routing Multi Copy* dengan menambahkan *Stationary Relay Node* dan *Management Buffer First In First Out* dengan ukuran *buffer* yang terbatas.

Dengan membatasi pada ukuran *buffer* ini akan menurunkan kinerja protokol *routing* dalam hal meningkatkan *delay* pengiriman dan menurunkan rasio pengiriman yang tidak ditujukan dalam hasil penelitian sebelumnya (IACC, 2016). Apabila protokol *routing Epidemic* mencapai rasio pengiriman optimal dengan *buffer* tak terbatas, skenario *buffer* yang terbatas dan kinerja *routing* terdegradasi mengakibatkan pelaksanaan *buffer* selalu penuh, sehingga banyak pesan yang tidak dapat dikirimkan. Untuk itu, agar dapat mengoptimalkan rasio pengiriman data dan memaksimalkan *buffer* pada setiap *node*, DTN dapat mengimplementasikan *Management Buffer Policy* sebagai pengoptimalan *buffer* yang terbatas saat pengiriman data.

Buffer Management Policy pada DTN merupakan suatu algoritma untuk mengatur pengelolaan pengiriman pesan, pesan mana yang akan di *drop* jika *Buffer* pada suatu *node* DTN penuh saat pesan baru akan diakomodasi di *node* tersebut (Filhoa, et al., 2016). Terlepas dari algoritma spesifik yang digunakan, sangat penting untuk memiliki kebijakan *drop* pada pesan yang efisien untuk mengoptimalkan penampaian pesan dalam sumberdaya yang terbatas (IACC, 2016). *First In - First Out* (FIFO) merupakan salah satu *simple management Buffer* yang mempunyai karakteristik pesan yang pertama datang akan diproses terlebih dahulu. Selain *Buffer Management Policy* untuk meningkatkan kinerja pada DTN terdapat mekanisme *Stationary Relay Node.*

Stationary Relay Node digunakan untuk meningkatkan probabilitas penerimaan pesan serta jembatan untuk pengiriman pesan. *Stationary Relay Node* merupakan sebuah *node* yang ditempatkan pada suatu tempat tertentu disepanjang jalur pengiriman data. *Node* bergerak yang melewati *Stationary Relay Node* dapat mengirimka salinan data yang belum terakomodasi serta dapat mengambil data pada *Stationary Relay Node* tersebut.

Pada peneltian kali ini diharapkan dapat memperoleh pengetahuan tentang DTN serta kinerja protokol *routing Multi Copy* menggunakan *Stationary Relay Node* dan *management buffer* FIFO dengan kondisi teknologi yang ada saat ini dan sumberdaya jaringan yang terbatas, sehingga dapat digunakan dalam pengiriman data digital dengan kondisi infrasturktur jaringan yang ekstrim secara nyata.

1.2 Rumusan masalah

Berdasarkan paparan latar belakang tersebut, maka rumusan masalah yang dapat dikaji adalah sebagai berikut:

1. Bagaimana merancang dan mensimulasikan teknologi DTN menggunakan protokol *routing Multi Copy* dengan *Stationary Relay Node* pada *The ONE Simulator*?
2. Bagaimana performansi *protocol routing* DTN *Multi Copy* dengan menambahkan *Management Buffer* menggunakan algoritma FIFO?

1.3 Tujuan

Adapun tujuan dari penelitian dan penulisan skripsi ini yaitu:

1. Merancang dan menganalisis teknologi DTN dengan protokol *routing Multi Copy* pada pengiriman data digital menggunakan *ONE Simulator*.
2. Mengetahui dan menganalisis performansi dari *routing* DTN *Multi Copy* menggunakan *Stationary Relay Node* dengan ditambahkan *Management Buffer* menggunakan algoritma FIFO.

1.4 Manfaat

Manfaat dari penelitian ini adalah sebagai media untuk memperoleh pengetahuan tentang DTN (*Delay Toleran Network*) serta kinerja dari protokol *routing Multi Copy* DTN dengan menambahkan *Stationary Relay Node* dan *Management Buffer* yang nanti hasilnya akan diperbandingkan untuk mengetahui *routing* mana yang paling optimal digunakan dengan kondisi teknologi yang ada saat ini yang mempunyai karakteristik sumberdaya jaringan yang terbatas, sehingga dapat digunakan dalam pengiriman data digital dengan kondisi infrasturktur jaringan yang ekstrim secara nyata.

1.5 Batasan masalah

Sesuai dengan masalah pada penelitian yang telah dipaparkan serta agar permasalahan yang dipaparkan lebih terfokus dan tidak meluas maka batasan-batasan yang ditentukan pada penilitian ini sebagai berikut:

1. Pembahasan difokuskan pada cara pengambilan data dari jaringan DTN dengan *routing Multi Copy* yaitu *Epidemic, ProPHET* dan *Spray And Wait*.
2. Penggunaan Teknologi *Opportunistic Network Environtment* (ONE) Simulator sebagai perancangan dan sebagai sarana pengambilan data uji.
3. Peta atau letek geografis pengujian pada *ONE Simulator* diperoleh dari *www.openstreetmap.org*
4. Parameter performansi yang digunakan dalam penelitian yakni *Delivery Probability* dan *Overhead Ratio.*
5. Batasan ukuran *buffer* yang digunakan adalah 30Mb, 50Mb, 80Mb dan 100Mb

6. Perancangan akan diterapkan dalam pengiriman data digital adalah sepanjang jalur dari area kebun percobaan Cangar Universitas Brawijaya sampai dengan Universitas Brawijaya Malang.
7. Mobile *Node* yang digunakan adalah berdasarkan trayek mobil sepanjang jalur dari Universitas Brawijaya ke daerah Cangar dan sebaliknya.
8. Pada pengujian dilakukan dua skenario pengujian yaitu dengan menggunakan *Single Destination* dan *Multiple Destination.*
9. Pada *Multiple Destination* pengaturan pada *event host* akan dibuat menjadi 2 *event* yang mempunyai kesamaan dengan *event host* pertama yaitu dengan *Events class* menggunakan *Message Event Generator* dan interval waktu yang digunakan setiap 35 detik serta ukuran pesan sebesar 5Mb tetapi menggunakan *node destination* yang berbeda.

1.6 Sistematika pembahasan

Penelitian ini diuraikan dengan sistemarika penulisan yang dibagi menjadi 7 bab pembahasan seperti berikut.

BAB I PENDAHULUAN

Pada bab ini berisi dasar dari pengangkatan masalah serta alasan penggunaan teknologi dan metode sebagai alat penyelesai dari masalah. Pada bab ini berisi pendahuluan berisi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB II LANDASAN KEPUSTAKAAN

Pada bab ini berisi tentang penjelasan singkat serta dasar teori dan referensi yang mendasari analisis dari sistem yang dibuat yang menjadi dasar penelitian rancang bangun pengiriman data digital pada DTN *routing* menggunakan *ONE Simulator*.

BAB III METODOLOGI

Berisi tentang metode-metode dan langkah-langkah kerja penelitian untuk analisa dan perancangan arsitektur pada *ONE Simulator*. Perancangna skenario bagaiamana penelitian ini dijalankan untuk mendapatkan hasil yang selanjutnya dapat di analisis

BAB IV PERANCANGAN SISTEM

Berisi tentang arsitektur sistem yang dibangun secara umum, struktur-struktur jaringan yang menjadi jembatan untuk pengiriman data. Konfigurasi – konfigurasi sistem yang perlu untuk dilakukan agar skekario yang telah dibuat dapat dijalankan dan di uji dalam sistem.

BAB V HASIL DAN ANALISIS

Pada bab ini membahas hasil skenario yang telah dibuat dalam pengirimkan data digital melalui jaringan dan topologi yang sudah dirancang sesuai dengan *routing* DTN yang diuji di dalam aplikasi *ONE Simulator*. Kemudian membahas tentang bagaimana performa dan skalabilitas dari sistem yang dibangun, pengujian data yang telah diambil dari analisa sistem yang telah dibuat, dan menganalisis hasil dari pengujian dengan parameter uji yaitu *Delivery Probability* dan *Overhead Ratio.*

BAB VI PENUTUP

Pada bab ini berisi kesimpulan yang dapat diambil berdasarkan hasil penelitian dan sistem yang telah dibangun serta saran terkait hal-hal yang perlu dikembangkan pada penelitian ini dan yang berguna bagi pengembangan sistem untuk kedepannya.

BAB 2 LANDASAN KEPUSTAKAAN

Pada pembahasan bab ini terdapat landasan teori yang merupakan kebutuhan sebagai dasar yang harus dipahamai dan dikaji agar pembaca bisa mengerti terkait dengan permasalahan yang diangkat dan dibahas dalam penelitian.

2.1 Kajian Pustaka

Table 2.1 Kajian Pustaka

No	Nama Penulis, Tahun dan Judul	Persamaan	Perbandingan	
			Penelitian Terdahulu	Rencana Penelitian
1	Poltak G. Hutajulu, 2017, Perbandingan Kinerja *Routing Multi Copy* dan *Routing First Contact* dengan *Stationary Relay Node* pada *Delay Tolerant Network* (DTN)	Menganalisis perbandingan *Routing* DTN *First Contact* dengan *Routing* DTN yang lain dan menambahkan *Stationary Relay Node*	Merancang dan menganalisis perbandingan *Routing* DTN *First Contact* dengan *Routing* DTN *Multi Copy* dan menambahkan *Stationary Relay Node* dengan mengasumsikan *buffer* tidak terbatas	Merancang dan menganalisis perbandingan *Routing* DTN *Multi Copy* dan menambahkan *Stationary Relay Node* serta *Management Buffer* untuk memenejemen *buffer* yang terbatas
2	Abdul Muis, 2013, Optimisasi Kinerja Manajemen *Buffer* Pada	Merancang dan menganalisis performansi *routing Multi copy Delay*	Merancang dan menganalisis performansi hanya *routing Multi copy Delay Tolerant Network*	Merancang dan menganalisis performansi *routing Multi copy Delay Tolerant*

	Jaringan *Delay Tolerant Network* (DTN) Untuk Jenis *Routing Multi Copy.*	*Tolerant Network* (DTN) dengan menambahkan manajemen *Buffer*	(DTN) dengan menambahkan manajemen *Buffer*	*Network* (DTN) dengan menambahkan *Stationary Relay Node* dan manajemen *Buffer* FIFO
3	Leanna VY, Jodi NR, 2016, Analisis Performansi Algoritma *Routing First Contact* dengan *Stationary Relay Node* pada *Delay Toleran Network*	Merancang dan menganalisis hasil dari kinerja *routing* DTN dengan *Stationary Relay Node*	Merancang dan menganalisis hasil kinerja dari satu *routing* DTN yaitu *First Contact* dengan *Stationary Relay Node*	Merancang dan menganalisis hasil kinerja dari *routing* DTN yaitu *Multi Copy* dengan *Stationary Relay Node*

2.2 Landasan Teori

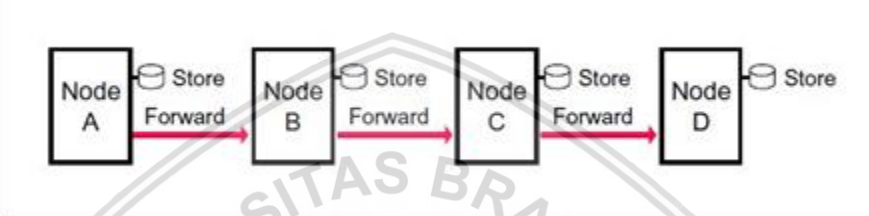
Berikut ini merupakan teori –teori yang digunakan dalam penelitian dalam membangun sistem didalam program *ONE Simulator* dan dengan menggunakan algoritma *routing* dari DTN *Multi Copy.* Adapun penjelasan-penjelasan teori tersebut seperti berikut:

2.2.1 *Delay Tolerant Network* (DTN)

Delay Tolerant Network (DTN) merupakan sebuah jaringan overlay yang beroperasi diatas berbagai jaringan regional, salah satunya seperti Internet. DTN memungkinkan jaringan regional dengan berbagai karakteristik *delay* untuk beroperasi dengan melibatkan *node – node* sebagai bagian struktur jaringan DTN untuk pengiriman data. Oleh karena itu, protokol yang mendasari dan teknologi untuk jaringan ini mungkin berbeda pada stiap arsitektur jaringannya, namun fleksibilitas dari arsitektur DTN memungkinkan teknologi tersebut dapat terhubung satu sama lain (LAP, 2010).

Delay Tolerant Network (DTN) adalah salah satu arsitektur end-to-end yang menyediakan komunikasi data dengan tingkatan *delay* variable yang besar, infrastruktur teknologi informasi yang minim, dan tingkat error yang tinggi. Untuk memberikan layanan, Bundle protocol merupakan lapisan penerapan di konstituen internet yang membentuk jaringan overlay store dan forward.

Pada DTN dikenal dengan metode simpan dan teruskan yaitu metode dimana semua data atau sebagian dari data *(fragment)* yang akan di kirim maka akan disimpan di sebuah *node* yang kemudian akan diteruskan ke *node* yang lain apabila telah terdapat jalur atau koneksi ke *node* yang lain sampai akhirnya data tersampaikan. Metode simpan data pada memory *node* dan teruskan ditunjukkan ke *node* berikutnya apa bila jalur tersedia seperti pada gambar 2.1.

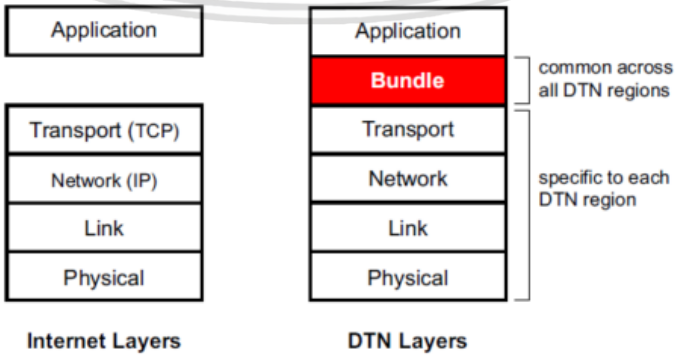


Gambar 2.1 Metode simpan dan teruskan pada DTN

Sumber : (Warthman, 2012)

2.2.2 Bundle Protocol

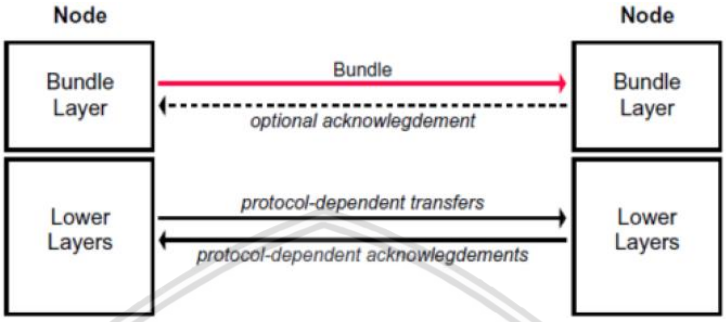
Bundle layer menyatukan lapisan – lapisan spesifik dari layer bagian bawah agar program atau aplikasi dapat berkomunikasi melalui lapisan yang berbeda-beda. Bundle merupakan data yang tersimpan sementara. Bundle layer menyimpan data dan meneruskannya keseluruhan pesan atau pecahan pesan (bundle fragments) dari satu *node* ke *node* yang lain. Sehingga data yang dikirmkan dari satu *node* ke *node* lain akan disimpan di *node* terakhir bila *node* selanjutnya memiliki koneksi yang terputus – putus. Sebuah lapisan bundle dapat digunakan di seluruh jaringan yang menggunakan teknologi DTN. Berikut adalah gambar bundle layer yang di tunjukan oleh gambar 2.2.



Gambar 2.2 Bundle Layer

Sumber : (Warthman F., 2012)

Pada jaringan yang koneksinya tidak selalu tersedia atau memiliki *delay* yang panjang, seperti protocol TCP yang membutuhkan pengenalan ACK akan mengakibatkan kegagalan untuk berkomunikasi. Oleh karena itu DTN menyediakan lapisan *bundle* yang akan saling berkomunikasi dengan lapisan *bundle* lainnya yang mempunyai sesi sederhana untuk memfalitasi pertukaran dan pengenalan ACK. Pertukaran ACK pada lapisan *bundle* ditunjukan pada gambar 2.3.



Gambar 2.3 Bundle Layer optional acknowlegdement

Sumber : (Warthman F., 2012)

2.2.3 Routing DTN *Multi Copy*

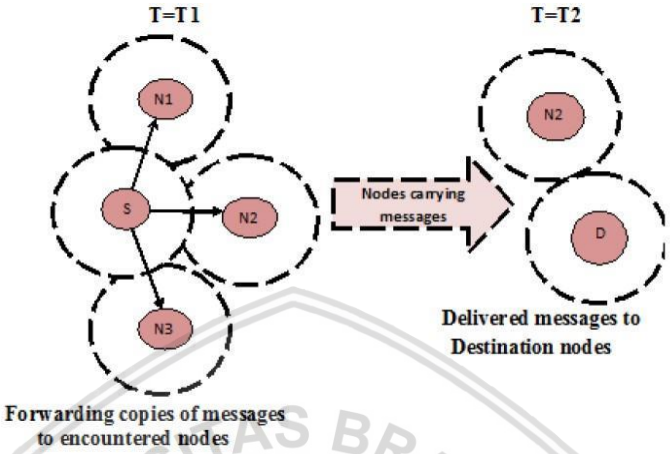
Pada jenis skema *routing Multi Copy* merupakan jenis protokol *routing* yang meneruskan setiap pesan pada node dibanyak jalur yang tersedia. Berbeda dengan mekanisme *Single Copy* yang hanya mengirimkan pesan unik ke jalur tunggal, sehingga dapat mengurangi kinerja jaringan berupa semakin meningkatnya penundaan dan rasio pengiriman namun sumberdaya jaringan yang dipakai tidak banyak. Dengan demikian dibandingkan dengan jenis *routing Single Copy*, jenis *routing Multi Copy* dapat memaksimalkan rasio pengiriman dan mengurangi penundaan tetapi membutuhkan tambahan sumber daya jaringan yaitu *energy, bandwith, storage* dan *buffer* (Muis, 2013). Ada beberapa *routing Multi Copy* diantaranya yaitu *Epidemic, SprayAndWait* dan *ProPHET*.

2.2.3.1 Routing Epidemic

Routing Epidemic merupakan *routing* yang mempunyai karakteristik *flooding-based forwarding* atau membanjiri jaringan dengan paket yang dikirimkan. Mekanisme pengiriman pada *Epidemic* yaitu dengan menyebarkan paket secara terus - menerus tanpa adanya batasan jumlah paket yang disebarkan pada *node* lain yang baru ditemukan dan node tersebut belum mempunyai salinan paket dari paket yang akan dikirimkan hingga masa TTL paket di jaringan berakhir. Pada *routing Epidemic* semua paket yang dikirimkan akan disebar keseluruh jalur dalam jaringan hingga paket sampai ke *node* tujuan. Tujuan dari protokol *routing Epidemic* ialah untuk meningkatkan rasio pengiriman pesan sampai tujuan dan memaksimalkan pesan *delay*. *Routing Epidemic* dapat menegaskan bahwa database tetap dapat distribusi dan dapat disinkronkan (Poltak, 2017)

Epidemic routing tidak sepenuhnya menjamin semua pesan yang dikirimkan dapat sampai ketujuan, namun *routing Epidemic* ini mampu membuat

jalur pendekatan terbaik untuk menyampaikan pesan ke *node* tujuan. *Routing Epidemic* memiliki kekurangan dalam pengiriman data yaitu keterbatasan *bandwith*, konsumsi energi, dan penggunaan memori karena saat mendistribusikan pesan, maka salinan pesan akan semakin banyak didalam jaringan. Ilustrasi pada pengiriman data *Epidemic* dapat di lihat pada gambar 2.4.



Gambar 2.4 Ilustrasi *routing Epidemic*

Sumber: (Poltak, 2017)

2.2.3.2 Routing ProPHET

Protokol *routing ProPHET* (*Probabilistic Routing Protocol Using History of Encounters and Transitivity*) adalah protokol yang bisa memprediksi probabilitas dengan menggunakan pengetahuan dari pertemuan sebelumnya dengan *node* lain untuk mengoptimalkan pengiriman paket. Setiap *node* menyimpan vektor perkiraan prediktabilitas pengiriman, dan menggunakannya untuk memutuskan apakah *node* yang ditemui adalah pembawa pesan ke tujuan.

Routing ProPHET merupakan hasil evolusi dari *routing Epidemic* (Poltak, 2017). Mekanisme pengiriman pesan pada *routing ProPHET* adalah mobilitas pengiriman *node* yang tidak sepenuhnya acak, akan tetapi memiliki sifat *deterministic* yaitu dengan mengulangi pola penyampaian pesan berdasarkan data vektor yang tersimpan, misalnya dalam *routing ProPHET* kemungkinan *node* telah bertemu dan mengujungi berbagai lokasi dalam beberapa waktu dapat meningkatkan data vektornya. Perbedaan signifikan *routing ProPHET* dengan *routing Epidemic* ialah strategi penyampaian pesan antar *node*, dimana *node* dengan data prediktabilitas vektornya terbaik akan diutamakan untuk jalur atau *node* pembawa pesan ke tujuan. Probabilitas data vektor pada *ProPHET* didefinisikan sebagai *Delivery Predictability.*

Untuk menghitung *Delivery Predictability* pada *ProPHET*, yang pertama ialah melakukan update terhadap data vektor prediktabilitas yang tersimpan oleh masing - masing *node*. Semakin sering *node* saling bertemu maka akan membuat data vektor prediktabilitas atau data *Delivery Predictability* yang tinggi. Berikut adalah cara menghitung *Delivery Predictability.*

Rumus mengupdate data *Delivery Predictability* Eq.2.1 dimana $P_{init} \in [0,1]$.

$$P_{(a,b)} = P_{(a,b)old} + (1 - P_{(a,b)old}) \, x \, P_{init} \quad \text{.....Eq.2.1}$$

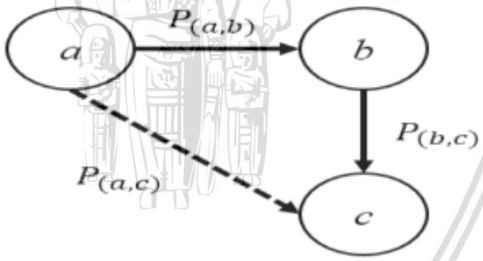
Apabila ada *node* yang sudah tidak pernah bertemu lagi satu sama lain. Maka *node* tersebut tidak lagi menjadi *node* yang mempunyai data *Delivery Predictability* yang baik dalam meneruskan pesan, sehingga nilai dari *Delvery Predictability*nya berkurang seiring dengan waktu seringnya pengiriman pesan dilakukan. Berikut merupakan rumus pengurangan (*aging*) Eq.2.2.

$$P_{(a,b)} = P_{(a,b)old} \, x \, \gamma^{k} \quad \text{.....Eq.2.2}$$

Dimana $\gamma \in [0,1]$ merupakan *aging constant* kemudian K adalah jumlah unit waktu yang telah terlewat semenjak terakhir kali data *Delivery Predictability* di update.

Delivery Predictability memiliki ciri *Transitive* yaitu jika *node* A sering bertemu dengan *node* B dan *node* B sering bertemu dengan *node* C, maka *node* C merupakan node yang bisa memiliki nilai *Delivery Predictability* yang baik untuk meneruskan pesan dari *node* A. Berikut merupakan rumus untuk *Transitivy* Eq.2.3.

$$P_{(a,c)} = P_{(a,c)old} + (1 - P_{(a,c)old}) \, x \, P_{(a,b)} \, x \, P_{(b,c)} \, x \, \beta \quad \text{.....Eq.2.3}$$



Gambar 2.5 Ilustrasi *Transitive* pada *routing Epidemic*

Sumber : (Poltak, 2017)

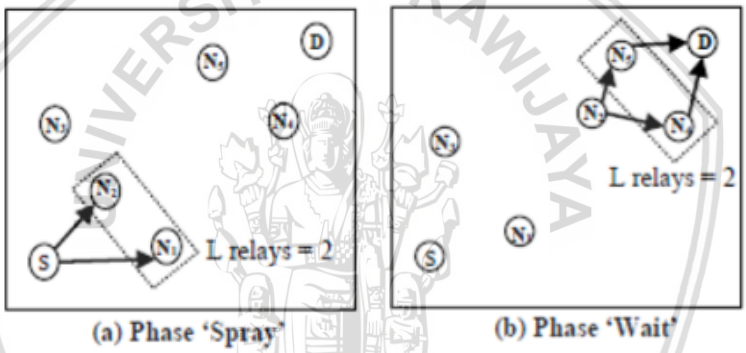
2.2.3.3 Routing Spray And Wait

Spray And Wait merupakan *routing* protokol dengan skema *routing* berbasis replikasi yang mencoba mendapatkan rasio pengiriman baik dengan memanfaatkan sumber daya rendah yang berbasis *routing forward based*. *Spray And Wait* memcapai efisiensi sumber daya dengan menetapkan batasan pada jumlah salinan per pesan dalam jaringan. *Spray and Wait* memiliki dua tahap metode, yaitu tahap *Spray* dan *Wait* (Wahanani, 2015).

Pada tahap *Spray node* sumber akan meneruskan salinan pesan ke *node* lain secara terus meneruh sepanjang jalur pengiriman akan tetapi pada setiap pesan yang didistribusian ke tujuan memiliki batasan dalam jumlah penyebarannya. Pesan yang disebarkan ini akan berhenti disebarkan apabila batasan tiap pesan

telah terpenuhi sesuai dengan jumlah salinan per pesan yang sudah ditentukan. Selanjutnya pada fase *Wait* penyebaran pesan pada jaringan dihentikan sesuai batasan jumlah salinan per pesan dalam jaringan. Pada fase *Wait* semua *node* yang membawa salinan pesan akan menunggu dan menuju langsung dengan *node* tujuan. setelah pesan sampai pada tujuan, maka konfirmasi pengiriman akan dilakukan dengan prinsip yang sama.

Strategi replikasi penyebaran pesan pada *routing Spray and Wait* mempunyai kesamaan dengan penyebaran pesan pada *routing Epidemic*. Gambar 2.5 pada bagian (a) menunjukkan fase *Spray* yaitu pesan yang dihasilkan oleh satu sumber pesan L akan disebarkan ke L *node relay* yang ditemuinya. Apabila pesan yang disebarkan belum mencapai ketujuan maka pesan akan didistibusikan lagi ke *relay node* terdekat sampai pesan sampai pada tujuan sebelum batasan jumlah salinan pesan habis atau masuk ke fase wait. Gambar 2.5 bagian (b) pada fase wait jika *node* tujuan tidak ditemukan pada fase *Spray*, maka masing – masing *node* yang membawa salinan pesan akan membawa pesan langsung ketujuan pengiriman.



Gambar 2.6 Ilustrasi fase *spray* dan fase *wait* pada pengiriman pesan *routing Spray And Wait.*

Sumber : (Poltak, 2017)

2.2.4 Stationary Relay Node

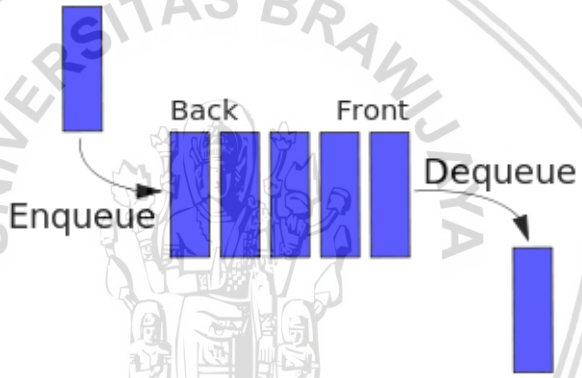
Stationary Relay Node merupakan *node* yang bertindak sebagai *node Relay* yang berperan untuk meneruskan pengiriman pesan ke *node* lainnya. *Stationary Relay Node* dapat meningkatkan probabilitas pertemuan antar *node* untuk bertukar pesan serta sebagai media transfer pesan menuju *node* tujuan (Leanna VY & NR, 2016; Mehto, 2013).

Stationary Relay Node merupakan perangkat Stationer yang mempunyai kemampuan *Store-and-forward* dan berada pada tempat yang telah di tentukan (Poltak, 2017). *Stationary Relay Node* mengijinkan *node* yang bergerak melintasinya serta dapat terkoneksi untuk mengambil data dan menyimpan data atau bertukar data satu sama lain. Penambahan pada *Stationary Relay Node* yang berguna sebagai Terminal data *node*, sehingga akan dapat meningkatkan *Delivery Probability* yang lebih tinggi (Spyropoulos, et al., 2004).

2.2.5 First In – First Out (FIFO)

Buffer Management Policy pada DTN merupakan suatu algoritma untuk mengatur pengelolaan pengiriman pesan, pesan mana yang akan di *drop* jika *Buffer* pada suatu *node* DTN penuh saat pesan baru akan diakomodasi di *node* tersebut (Filhoa, et al., 2016). Terlepas dari algoritma spesifik yang di gunakan, sangat penting untuk memiliki kebijakan *drop* pada pesan yang efisien untuk mengoptimalkan penampaian pesan dalam sumberdaya yang terbatas (IACC, 2016). *First In - First Out* (FIFO) merupakan salah satu *simple management Buffer* yang mempunyai karakteristik pesan yang pertama datang akan di proses terlebih dahulu.

First In – First Out (FIFO) merupakan metode untuk mengatur dan memanipulasi *buffer* data, dimana entri pertama atau kepala akan di proses terlebih dahulu dan pada antrian tersebut pesan yang datang pertama pula yang akan di jatuhkan terlebih dahulu. Pemodelan antrian pada algoritma FIFO dapat dilihat pada gambar 2.7.



Gambar 2.7 Pemodelan antrian pada algoritma FIFO

Sumber : (Fajri, 2016)

Penjadwalan dan pesan yang di *drop* pada FIFO memudahkan *node* untuk menyimpan *bundle fragment* dan mengakomodasi pesan baru pada pengiriman pesan DTN. Berikut adalah tabel skanario penjadwalan dan penjelasan dari algoritma penjadwalan FIFO dalam *memanagement buffer fragment* dapat dilihat pada tabel 2.1.

Frame	2	3	1	4	7	1	2	5	6	2
0	2	2	2	4	4		4	5	5	
1		3	3	3	7		7	7	6	
2			1	1	1	Skip	2	2	2	Skip

Tabel 2.1 Skenario penjadwalan FIFO dalam *memanagement buffer*

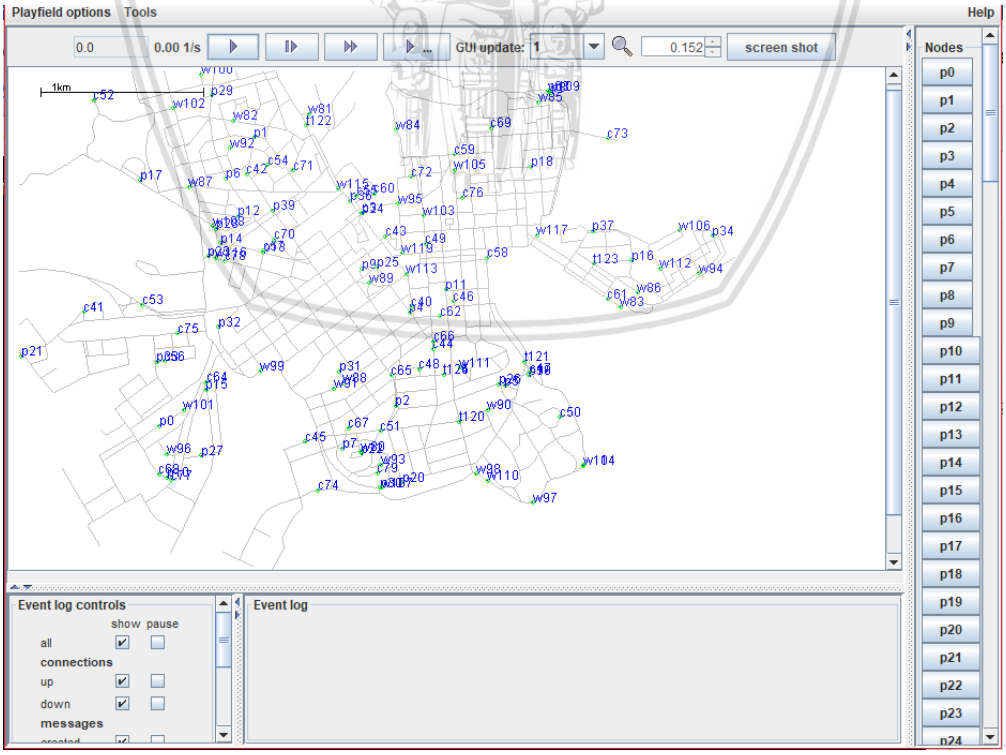
Diasumsikan terdapat 3 frame pada *buffer node*, pada data diatas diketahui ada 10 paket fragment yang akan di kirimkan, fragment dengan angka 2, 3 dan 1 menempati frame *buffer* yang masih belum terisi yaitu 0, 1 dan 2, kemudian saat pesan fragment 4 akan di akomodasikan ke *buffer node* maka fragment 2 akan

dihapus terlebih dahulu karena menempati memory paling lama, ini dikarenakan mekanisme dari FIFO yang diterapkan pada *buffer management.* Kemudian fragment 3 di*drop* untuk mengakomodasi fragment 7 ke dalam memory buffer. Pada fragment 1 dikarenakan dalam satu frame yang diakomodasi dalam satu waktu fragment 1 di skip karena fragment 1 masih ada dalam penyimpanan *buffer*. Begitu seterusnya sampai data berhasil sampai tujuan atau waktu TTL pesan yang di kirim habis didalam jaringan.

2.2.6 ONE Simulator

Simulator *Opportunistic Network Environtment* (ONE) adalah simulator yang digunakan untuk merancang dan mengevaluasi *routing* pada jaringan *Opportunistic* atau DTN. *ONE simulator* mempunyai fungsi utama memodelkan hubungan antar *node*, pergerakan dari *node*, model *routing* dan penanganan dalam pengiriman serta penerimaan pesan. Dari pemodelan yang di buat pada *ONE simulator* hasil yang akan diperoleh melalui visualisasi serta *post processing tools* (Widhiyanto, 2016).

ONE simulator dibangun menggunakan bahasa pemrograman *Java.* Dalam *ONE simulator* ini pemodelan dan pergerakan digunakan sebagai monitoring jika dua *node* atau lebih dapat berkomunikasi untuk melakukan pertukaran informasi (Jodi Nugroho Restu, 2015). Pemodelan map, pergerakan *node* dan simulasi perutean dapat diamati pada GUI simulator dan salah satu fasilitas dari *ONE Simulator* yaitu *Report Module* untuk mengetahui informasi dari proses simulasi. Berikut gambaran dari aplikasi *ONE Simulator* dapat di lihat pada gambar 2.8.



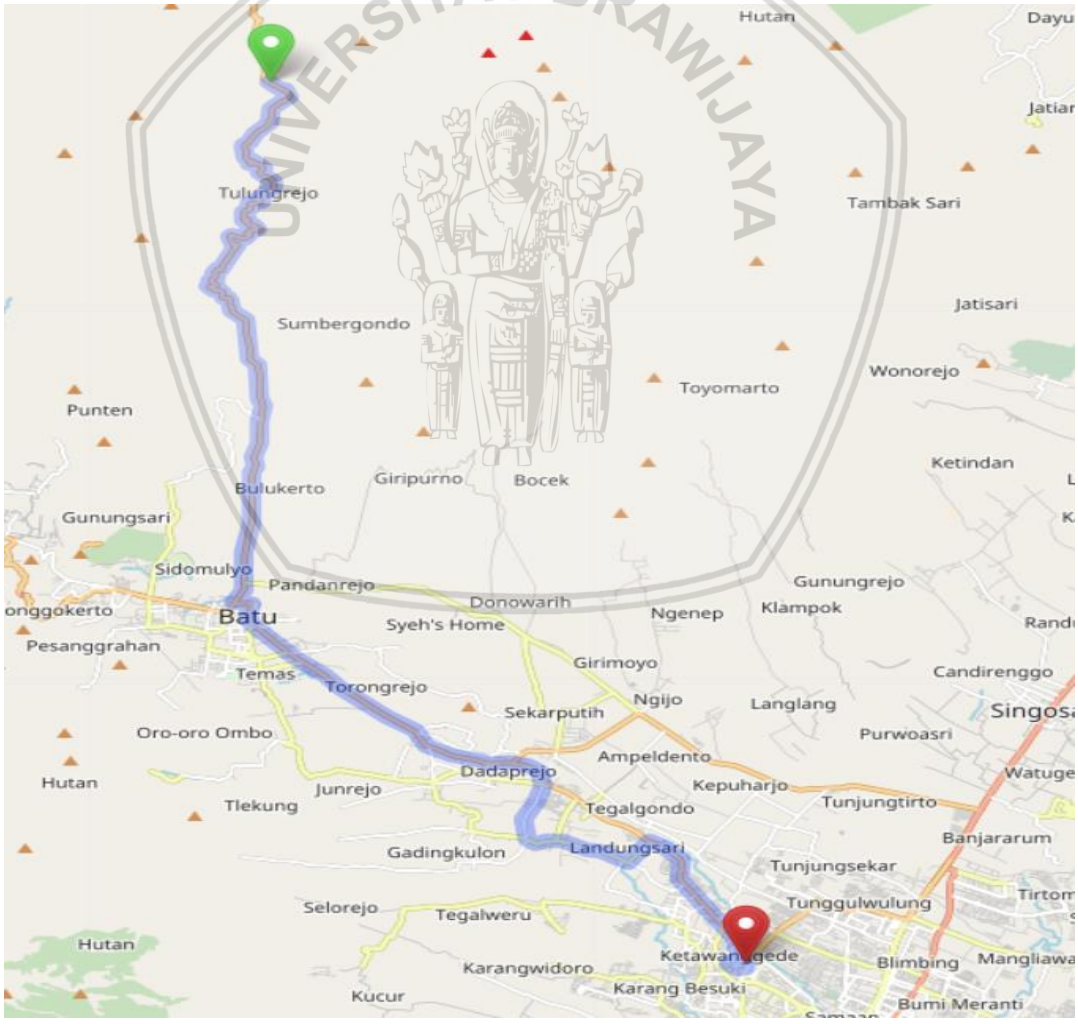
Gambar 2.8 *GUI ONE simulator*

Sumber : (Poltak, 2017)

2.2.7 OpenStree Map (OSM)

Open Street Map (OSM) merupaka sebuah proyek yang berbasis website untuk membuat peta dunia secara *free* atau gratis dan terbuka untuk siapa saja. OSM dirintis oleh Steve Coast di Britania pada tahun 2004. OSM dikembangkan dan dibangun oleh sukarelawan dengan cara mensurve menggunakan GPS, citra digital dari satelit dan mengumpulkan data dari geografis yang tersedia di publik (Restu & Yovita, 2015).

Pada penelitian ini OSM digunakan untuk membuat peta geografis dan rute sebagai bahan untuk pemodelan penelitian pada *ONE Simulator*. Peta geografis yang di ambil pada area kebun percobaan Cangar Universitas Brawijaya yang berada di Cangar, Batu Malang Jawa Timur, dengan letak koordinat -7.864085, 112.361037 sampai dengan Universitas Brawijaya Malang yang berada di jalan Veteran, Kecamatan Lowokwaru, Ketawanggede, kota Malang dengan koordinat -7.9371978, 112.5990894. Berikut adalah gambaran map dari *Open Street Map* (OSM) dapat di lihat pada gambar 2.9.



Gambar 2.9 Tampilan peta dari jalur area kebun percobaan Cangar Universitas Brawijaya sampai dengan Universitas Brawijaya Malang

2.2.8 Parameter Performansi

Parameter performansi adalah alat ukur pengujian berupa rumus untuk menganalisis hasil dari skenario yang telah dijalankan, sehingga diperoleh informasi apa yang sedang terjadi pada jaringan tersebut. Ada bermacam – macam parameter performansi yang digunakan dalam jaringan. Dalam penelitian ini parameter performansi yang digunakan ialah *Delivery Probability* dan *Overhead Ratio* sebagai alat ukur pengujian yang selanjutnya akan dianalisis hasilnya.

2.2.8.1 Delivery Probability

Delivery Probability adalah perbandingan antara rasio dari banyaknya jumlah total pesan yang dikirim ke tujuan dengan jumlah pesan yang dibuat pada *node* sumber yang dikirimkan dalam suatu periode waktu yang telah di tentukan (Gamit & Patel, 2014). *Delivery Probability* menunjukan tingkat keberhasilah sebuah protokol *routing* dalam mendistribusikan pesan ke tujuan. Apabila nilai dari *Delivery Probability* semakin tinggi, maka pesan yang sampai ke tujuan akan semakin banyak. Berikut ini adalah rumus untuk mencari nilai dari *Delivery Probability* Eq.2.4.

$$\textit{Delivery Probability} = \frac{\mathrm{D}}{\mathrm{G}} \quad \text{.....Eq.2.4}$$

Keterangan :

D = jumlah pesan yang sampai ke tujuan

G = jumlah pesan yang dibuat dari *node* sumber

Sumber : (Mehto, 2013)

2.2.8.2 Overhead Ratio

Overhead Rotio merupakan berapa banyak paket redudansi terkait untuk peiriman satu pesan ke tujuan *Overhead Rotio* dipengaruhi oleh waktu yang digunakan sebuah pesan dalam *buffer* sebelum pesan diteruskan ke *node* yang lain. Apabila nilai dari *Overhead Rotio* semakin rendah maka semakin optimal protokol *routing* yang digunakan. Berikut adalah rumus untuk mencari nilai dari *Overhead Ratio* Eq.2.5.

$$\textit{Overhead Ratio} = \frac{R-D}{\mathrm{D}} \quad \text{.....Eq.2.5}$$

Keterangan :

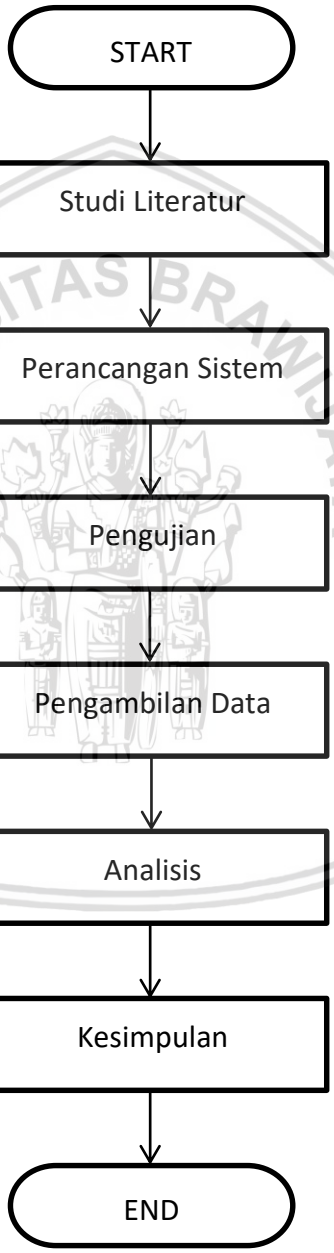
D = jumlah pesan yang sampai pada tujuan

R = jumlah pesan yang di teruskan oleh *node*

Sumber : (Mehto, 2013)

BAB 3 METODOLOGI

Bab ini menjelaskan langkah – langkah dalam pembuatan sistem. Metodologi penelitian yang dilakukan dalam penelitian ini melalui beberapa tahapan yaitu studi literature, perancangan sistem, pengujian, pengambilan data, analisis dan kesimpulan. Berikut langkah - langkah pengerjaan sistem yang digambarkan dalam diagram blok metode penelitian.



3.1 Studi Literatur

Studi literatur merupakan penjelasan tentang beberapa *paper* yang terkait dengan penelitian ini, teori pendukung yang diperoleh dari jurnal, makalah ilmiah, dan beberapa penelitian sebelumnya yang terkait dengan penulisan penelitian yang akan di lakukan oleh peneliti. Studi literatur yang ada pada penelitian ini yaitu mengenai *Routing* protokol *Delay toleran network Multi Copy* dan *Management Buffer First In First Out* serta *Stationary Relay Node* dengan menggunakan aplikasi *ONE Simulator.*

3.1.1 Analisis Kebutuhan

Pada analisis kebutuhan peneliti membuat analisis kebutuhan sistem yang di bagi menjadi dua yaitu kebutuhan perangkat keras dan kebutuhan perangkat lunak. Berikut adalah spesifikasi kebutuhannya.

A. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang digunakan dalam penelitian ini mempunyai sepesifikasi perangkat keras yang mampu mendorong dan menjalankan berbagai macam aplikasi serta multi tasking. Berikut adalah spesifikasi perangkat keras yang digunakan dalam pengerjaan tugas akhir ini :

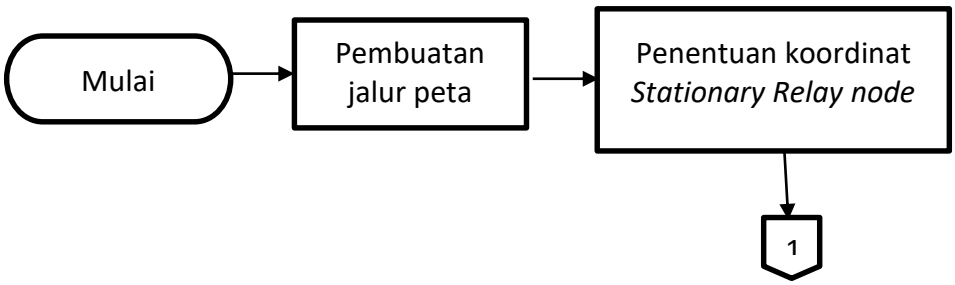
- Laptop : Asus X550UI
- RAM : 8.0 Gb
- Processor : Intel(R) Core i7-4700HQ CPU @2.40GHz
- Sistem operasi : Windows 10 Home 64-bit

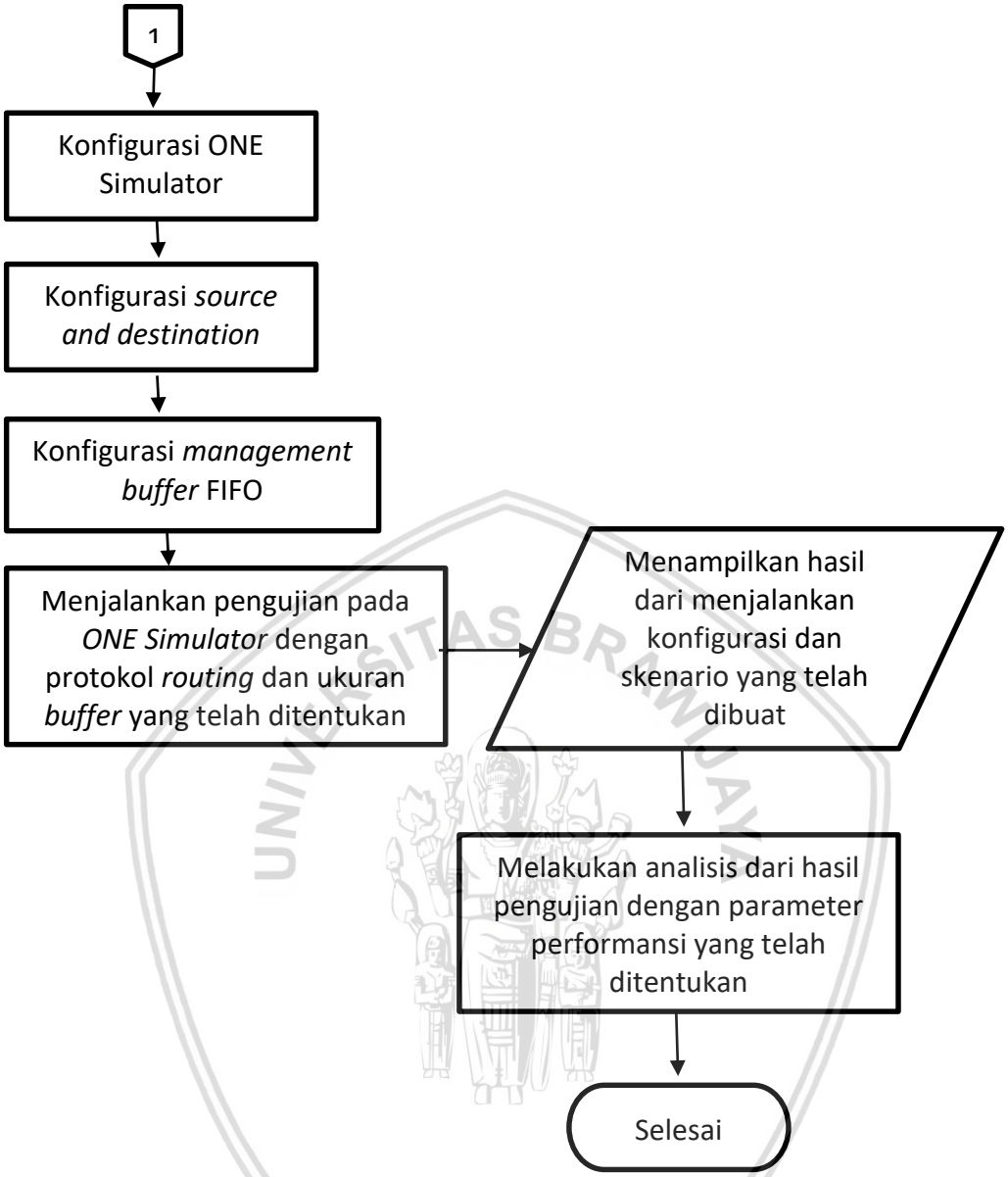
B. Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan untuk pengerjaan penelitian ini meliputi aplikasi *ONE Simulator* versi 1.6.1. Selain *ONE Simulator*, aplikasi lain yang di butuhkan adalah *Open Jump* untuk mencari kordinat sebagai kordinat penempatan *Stationary Relay Node,* kemudian Google Chrome untuk mengakses *Open Street Map* (OSM) dan mengambil peta geografi sebagai bahan pengerjaan tugas akhir ini.

3.2 Perancangan Sistem

Rancangan skenario sistem meliputi arsitektur dari sistem, gambaran dari sistem yang dibuat secara umum. Rancangan scenario sistem merupakan identifikasi dari analisi kebutuhan yang di lakukan untuk menentukan system dan infrastruktur yang di bangun sesuai dengan kebutuhan pengiriman data. Menentukan kebutuhan yang di perlukan sistem perancangan infrastruktur bebasih *Delay Tolerant Network* sebagai media tranmisi datanya.





Gambar 3.1 *Flowchart* perancangan sistem

Pada gambar 3.1 merupakan *Flowchart* perancangan dari sistem yang akan diujikan dalam penelitian ini. Perancangan yang dibuat meliputi pembuatan jalur peta untuk jalur pergerakan *node,* penentuan koordinat *Stationary Relay Node* untuk menentukan letak *Stationary Relay Node* sepanjang jalur pengiriman pesan. Kemudian konfigurasi *ONE Simulator* yang meliputi instalasi *ONE Simulator,* konfigurasi waktu skenario, pergerakan *node,* penentuan protokol *routing,* batasan ukuran *buffer,* penentuan *Time To Live* dari pesan yang didistribusikan, dan lain – lain. Selanjutnya melakukan konfigurasi *source and destination* yaitu menentukan *node source* dan *node destination* dengan skenario *single destination* dan *multi destination.* Kemudian melakukan konfigurasi *management buffer* dengan algoritma *first in – first out.* Setelah itu konfigurasi dijalankan pada *ONE*

Simulator dan dihasilkan data yang selanjutnya dilakukan analisis dengan parameter yang telah ditentukan.

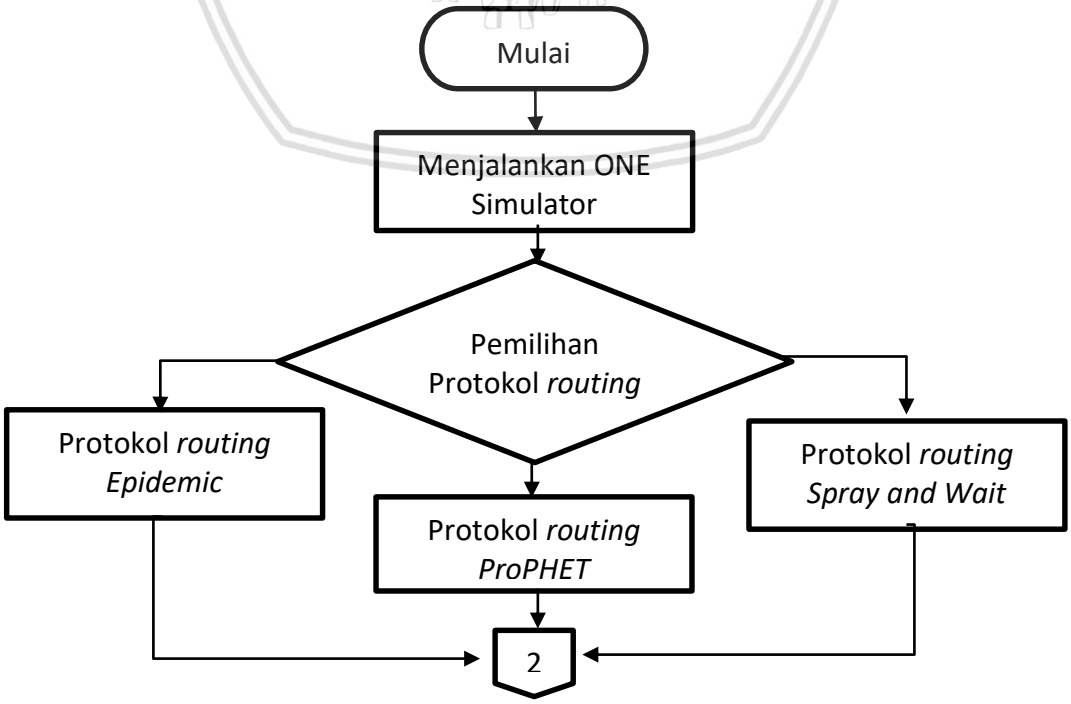
Peta geografis yang di gunakan menggunakan jalur mobil dari area kebun percobaan Cangar Universitas Brawijaya sampai dengan Universitas Brawijaya Malang. Perancangan *node* dan *Stationary Rela*y *Node* yang di gunakan sebagai *node* untuk menjembatani data agar data dapat dikirimkan ke tujuan. *Node* dengan jumlah tertentu akan tersebar dan secara acak bergerak dari wilayah area kebun percobaan Cangar Universitas Brawijaya sampai dengan Universitas Brawijaya. *Stationary Relay Node* akan diletakan secara acak.

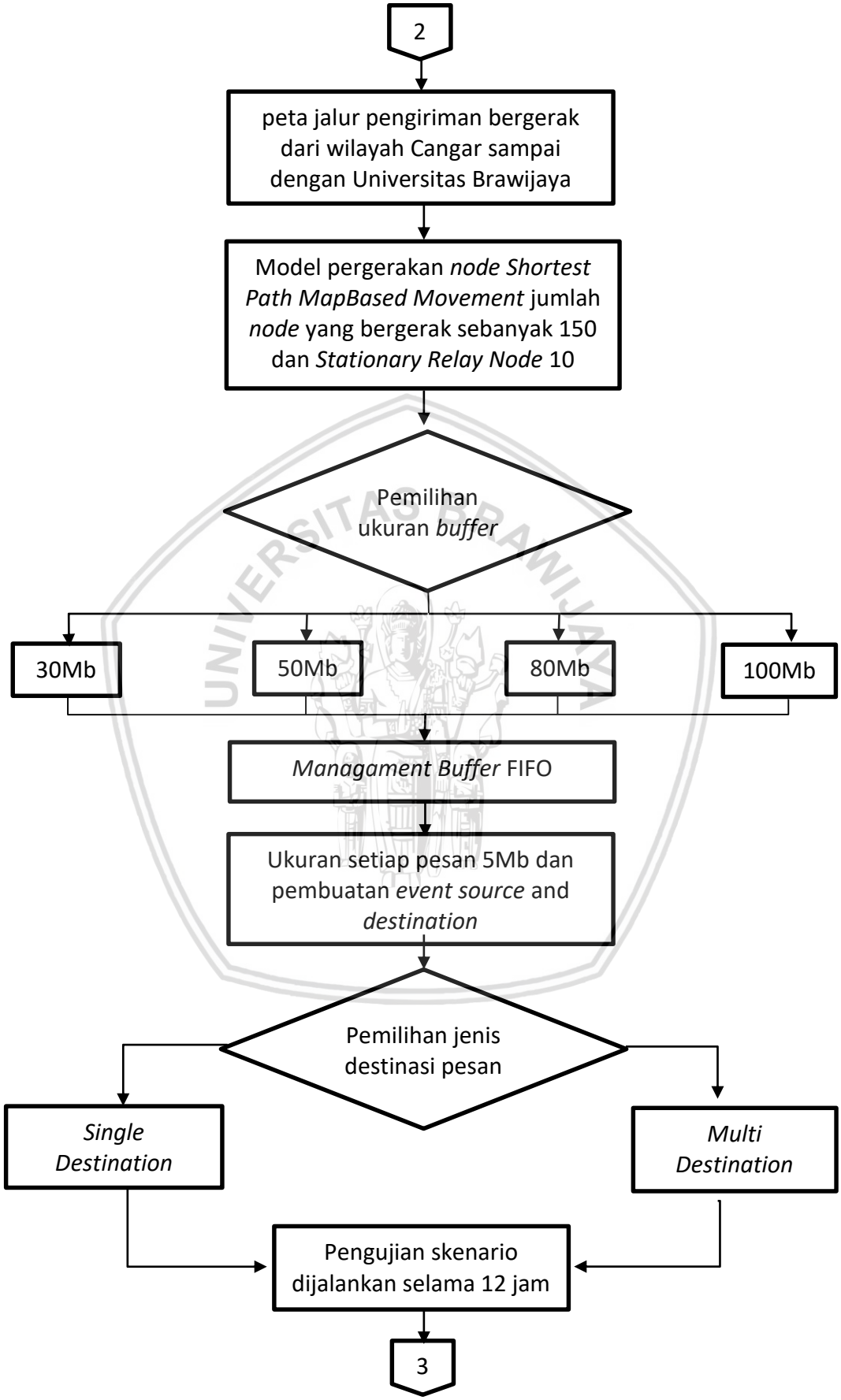
Management Buffer pada *node* menggunakan algoritma *First In - First Out* (FIFO). Sehingga pengelolaan *buffer* pada *node* dapat di maksimalkan untuk pengiriman dan penerimaan pesan pada *node*. FIFO memenejemen *buffer* dengan cara melayani pesan atau paket yang masuk terlebih dahulu, dan bagaimana menjatuhkan pesan apa bila pesan baru yang datang akan di akomodasikan ke dalam buffer *node*.

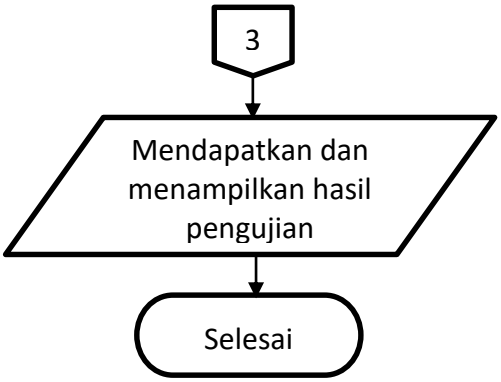
Dalam pengiriman data jalur yang digunakan ialah area kebun percobaan Cangar Universitas Brawijaya sampai dengan Universitas Brawijaya Malang, system yang dirancang menggunakan protokol *routing* DTN *Multi copy*.

3.3 Pengujian

Pada tahap ini dari perancangan sistem yang telah dibuat, maka akan dilakukan pengujian menggunakan Simulator *Opportunistic Network Environtment* (ONE). *ONE simulator* digunakan sebagai alat uji untuk mendapatkan hasil dan data yang nantinya akan dianalisis dalam pengambilan kesimpulan pada tugas akhir ini. Berikut adalah skenario simulasi yang akan dilakukan:




2
peta jalur pengiriman bergerak dari wilayah Cangar sampai dengan Universitas Brawijaya
Model pergerakan *node Shortest Path MapBased Movement* jumlah *node* yang bergerak sebanyak 150 dan *Stationary Relay Node* 10
Pemilihan ukuran *buffer*
30Mb
50Mb
80Mb
100Mb
Managament Buffer FIFO
Ukuran setiap pesan 5Mb dan pembuatan *event source* and *destination*
Pemilihan jenis destinasi pesan
Single Destination
Multi Destination
Pengujian skenario dijalankan selama 12 jam
3



Gambar 3.2 *Flowchart* pengujian skenario dari sistem yang telah dibuat

Gambar *Flowchart* 3.2 diatas menunjukan skenario pengujian yang akan dijalankan pada penelitian ini. Berikut adalah penjelasan skenario pengujian yang akan dilakukan.

1. Pada skenario pengujian yang menggunakan *routing Epidemic*, simulasi dijalankan menggunakan ONE Simulator protokol *routing* yang gunakan adalah *Epidemic* dan penggunaan *Stationary Relay node* sebanyak 10 buah yang telah di tempatkan pada area tertentu dan *node* yang bergerak sebanyak 150 *node*, serta menggunakan *management buffer* FIFO dengan batasan ukuran *buffer* sebesar 30Mb, 50Mb, 80Mb dan 100Mb. Pesan yang dibuat memiliki ukuran sebesar 5Mb. Pengujian dijalankan menggunakan skenario *Single Destination* dan *Multi Destination.*
2. Pada skenario pengujian yang menggunakan *routing ProPHET*, simulasi dijalankan menggunakan ONE Simulator protokol *routing* yang gunakan adalah *ProPHET* dan penggunaan *Stationary Relay node* sebanyak 10 buah yang telah di tempatkan pada area tertentu dan *node* yang bergerak sebanyak 150 *node*, serta menggunakan *management buffer* FIFO dengan batasan ukuran *buffer* sebesar 30Mb, 50Mb, 80Mb dan 100Mb. Pesan yang dibuat memiliki ukuran sebesar 5Mb. Pengujian dijalankan menggunakan skenario *Single Destination* dan *Multi Destination.*
3. Pada skenario pengujian yang menggunakan *routing Spray and Wait*, simulasi dijalankan menggunakan ONE Simulator protokol *routing* yang gunakan adalah *Spray and Wait* dan penggunaan *Stationary Relay node* sebanyak 10 buah yang telah di tempatkan pada area tertentu dan *node* yang bergerak sebanyak 150 *node*, serta menggunakan *management buffer* FIFO dengan batasan ukuran *buffer* sebesar 30Mb, 50Mb, 80Mb dan 100Mb. Pesan yang dibuat memiliki ukuran sebesar 5Mb. Pengujian dijalankan menggunakan skenario *Single Destination* dan *Multi Destination.*

Pada skenario pengujian masing – masing skenario dijalankan sesuai dengan rute dari wilayah area Cangar sampai dengan Universitas Brawijaya Malang dengan kecepatan *node* berjalan 10 – 50 Km/jam. Waktu yang digunakan untuk simulasi selama 12 jam dengan besar ukuran pesan yang dikirim yakni 5Mb dan dengan batasan *Buffer* pada semua *node* yakni 30Mb, 50Mb, 80Mb dan 100Mb.

3.4 Pengambilan Data

Pada tahap ini data uji yang diambil dan dikumpulkan dilakukan saat pengujian telah selesai, sehingga pengambilan data dari hasil pengujian pada simulator yang telah telah terkumpul dapat dilakukan perhitungan yang menjadi data hasil. Kemudian data hasil dari akan dimasukan kedalam tabel data yang nantinya akan diubah menjadi data statik, data hasil diperoleh dengan perhitungan menggunakan parameter performansi.

Pada penelitian ini parameter performansi yang digunakan ialah *Delivery Probability* dan *Overhead Ratio* sebagai alat ukur pengujian dan analisinya. Setalah didapatkan nilai *Delivery Probability* dan *Overhead Ratio* tahap selanjutnya adalah menganalisa data hasil yang sudah diubah menjadi data statik dan dilakukan pengambilan kesimpulan.

3.5 Analisis

Pada bagian analisis ini menjelaskan perbandingan dari model *routing* yang telah di uji pada simulator dengan skenario yang telah ditentukan. Kemudian hasilnya dari masing-maring *routing* di bandingkan satu dengan yang lain. Selanjutnya perbandingan *routing* yang diuji akan didapatkan hasil protokol *routing* yang terbaik dalam perbandingan kinerja dan pengiriman data.

3.6 Kesimpulan

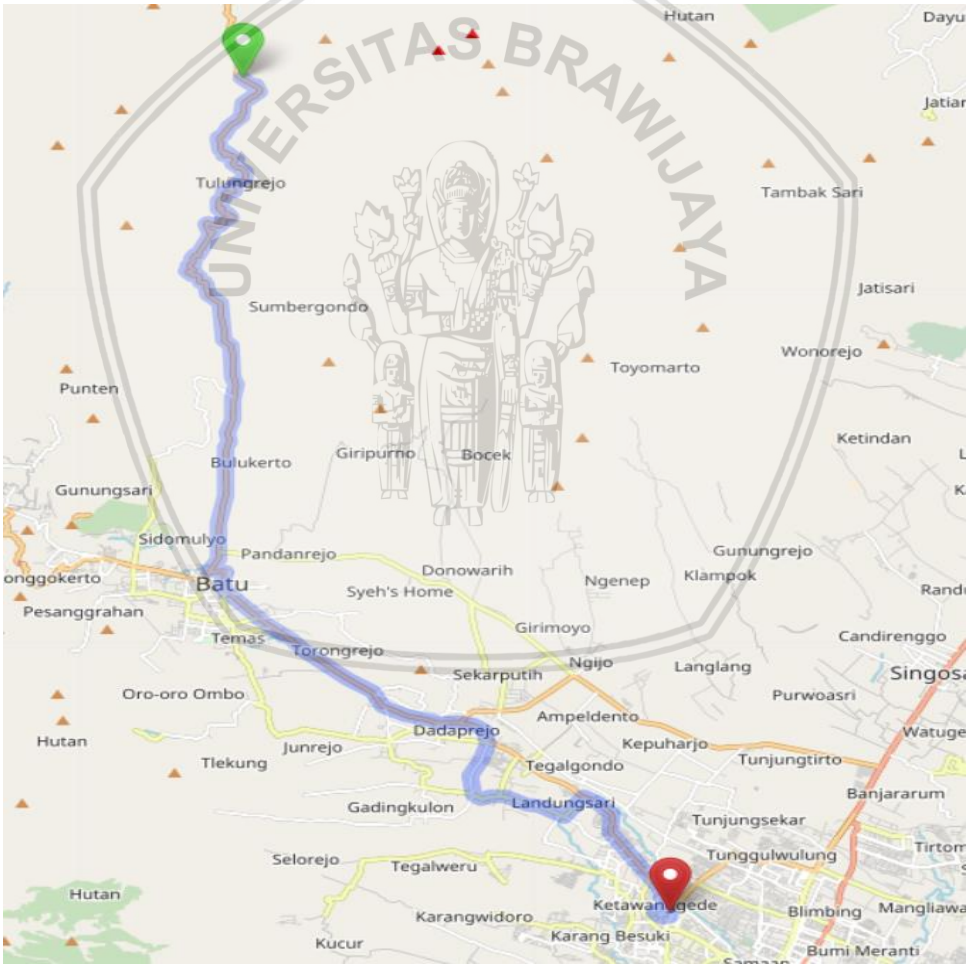
Dari hasil pengujian dan analisis, dilakukan pengambilan atau penarikan kesimpulan dari tahapan tersebut tentang system yang dapat terintegrasi beserta presentasi keberhasilan dari system secara utuh. Kemudian akan dilakukan penarikan kesimpulan dari kinerja masing - masing *routing* dalam scenario uji yang telah di buat.

BAB 4 PERANCANGAN SISTEM

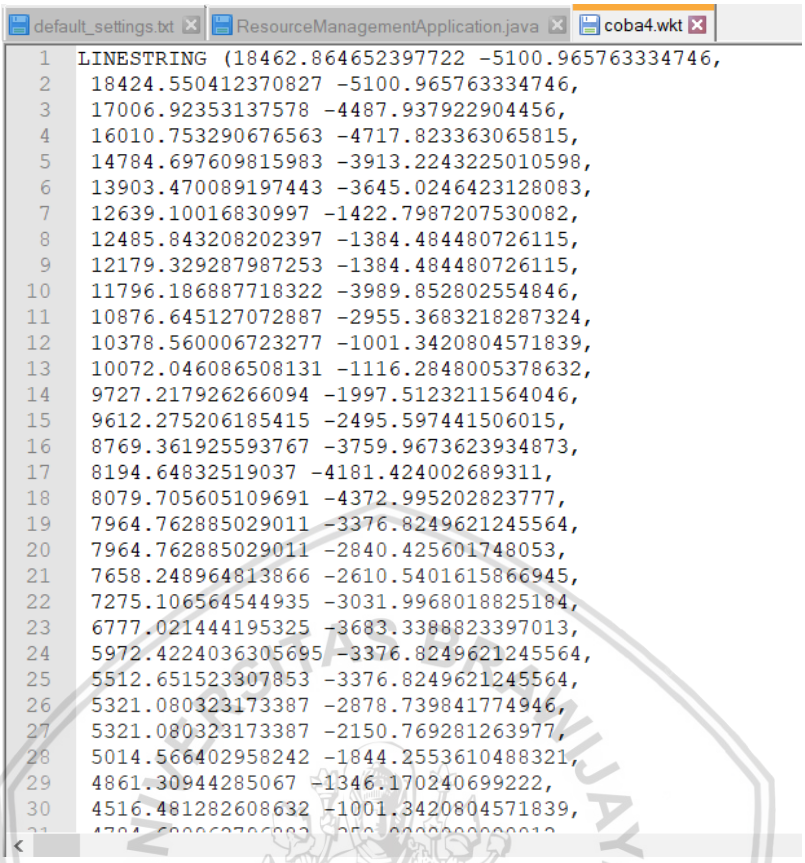
Pada bab ini mejelaskan tahapan – tahapan merancang sistem dalam pengujian kinerja pada protokol *routing Multi copy* dengan penambahan *Stationary relay node* dan *management buffer* dengan algoritma *First in - First Out* sesuai dengan perancangan yang akan di buat.

4.1 Pembuatan Rute Pada *Open Steet Map*

Pada pembuatan rute pengirima pesan didapatkan dari website www.openstreetmap.org dengan mengambil jalur kendaraan dari area kebun percobaan Cangar Universitas Brawijaya sampai dengan Universitas Brawijaya Malang dengan format .wkt (*well know text*). Selanjutnya data di olah menggunakan aplikasi *Open jump* untuk membuat kordianat sebagai tempat untuk menempatkan *Stationary relay node*.



Gambar 4.1 Tampilan jalur dari area kebun percobaan Cangar Universitas Brawijaya sampai dengan Universitas Brawijaya Malang



Gambar 4.2 Bentuk jalur yang akan di gunakan dalam pertukaran pesan dalam bentuk .wkt

Gambar 4.2 merupakan *convert* dari peta yang ada pada *Open Street Map* dengan batas yang sudah di tentukan, kemudian file tersebut di simpan dalam bentuk .wkt yang dapat dibaca oleh *ONE Simulator*.

4.2 Konfigurasi ONE Simulator

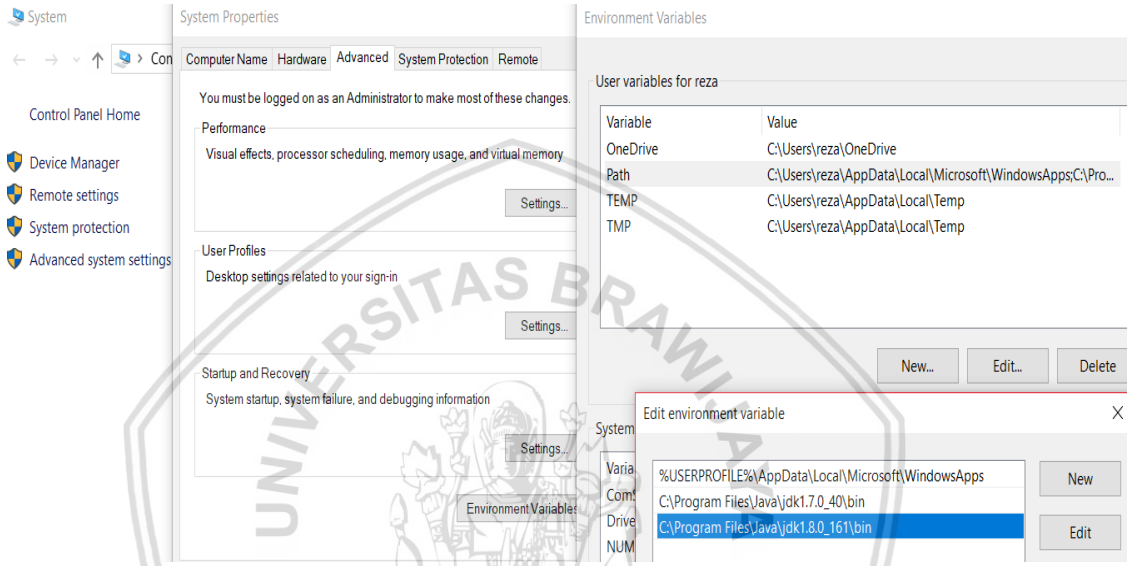
4.2.1 Instalasi ONE Simulator

Untuk mendapatkan *ONE Simulator* dapat di *download* pada website *http://akeranen.github.io/the-one/.* Terdapat beberapa versi untuk *ONE Simulator* dari pengembangan yang telah terjadi. Pada penelitian ini menggunakan *ONE Simulator* versi 1.6.0 yang di jalankan pada sistem operasi *windows* 10.

4.2.2 Setting Java Path Variable

Untuk menjalankan *The ONE Simulator,* terlebih dahulu kita men-*download* dan memasang file *JDK (Java Development Kit)* pada sistem yang akan digunakan. JDK merupakan *software* yang digunakan sebagai proses kopilasi dari kode java ke *bytecode* yang dapat dibaca dan dijlankan oleh JRE (*Java Runtime Environment*). Setelah JDK di *download* kemudian menginstal JDK dan selanjutnya melakukan pengaturan pada *Java Path* di *windows Environment Variable.* Berikut adalah cara pengaturan *Java Path Variable* pada sistem windows 10 :

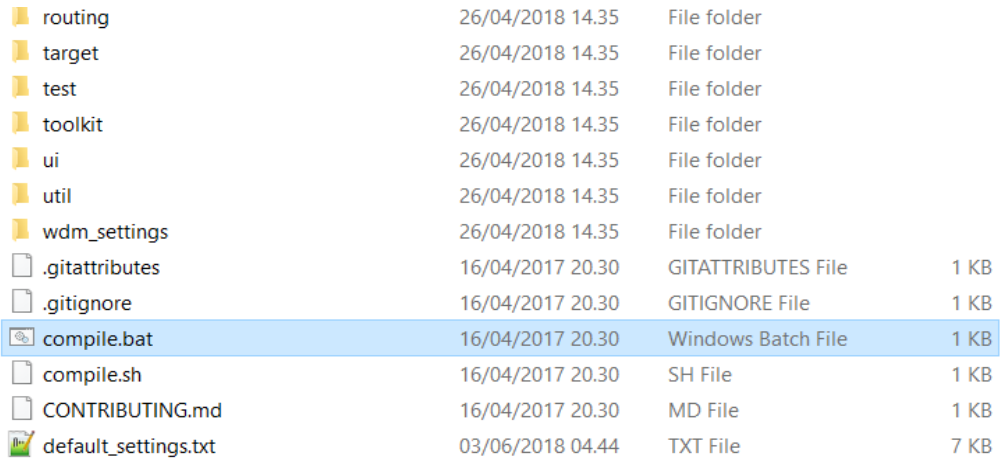
1. Masuk ke *My Computer* dengan melakukan klik kanan *icon This PC* pada program *explorer* dan memilih *properties*,
2. Berikutnya klik pada *Advanced System Setting* dan pilih *System Variable,*
3. Selanjutnya pilih *path* kemudian klik *edit* untuk memasukan folder JDK yang telah terinstal. Menambahkan *path* folder JDK1.8.0 dengan cara memasukan tulisan letak folder JDK seperti berikut. *C:\Program File\Java\jdk1.8.0_161\bin*. Langkah – langkah dari *Setting Java Path Variable* padat di lihat pada gambar 4.3.



Gambar 4.3 Langkah – langkah *Setting Java Path Variable*

4.2.3 Compile dan Runing The ONE Simulator

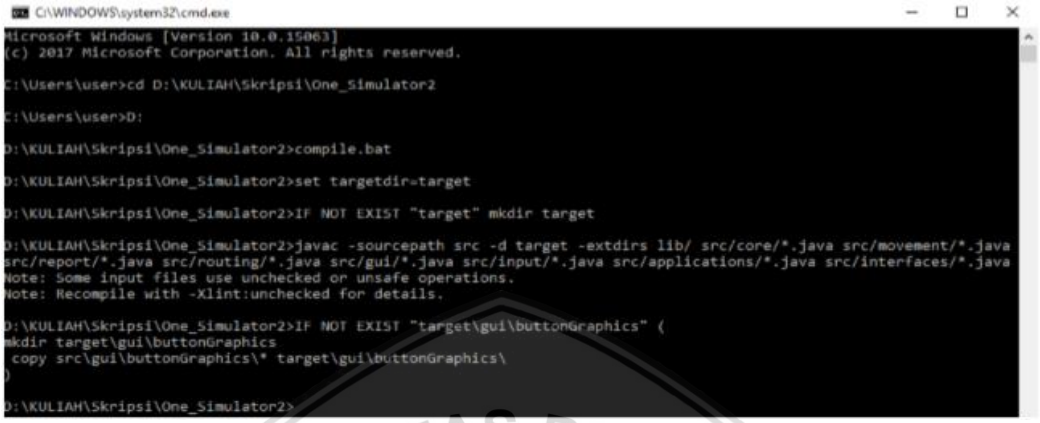
Setelah melakukan *Setting Java Path Variable,* selanjutnya melakukan *Compile* pada *The ONE Simulator.* Dengan melakukan *Compile* maka *ONE Simulator* akan dapat di jalankan. Untuk melakukan *Compile* pertama mencari file dengan nama *compile.bat* pada folder *ONE Simulator* seperti gambar 4.4 berikut.



Gambar 4.4 File *compile.bat* pada folder *ONE Simulator*

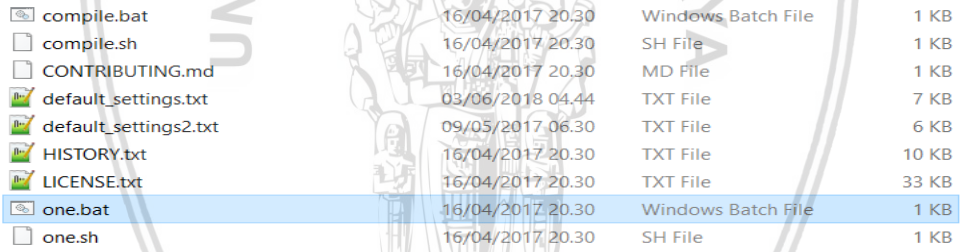
Untuk menjalankan compile dapat menggunakan 2 cara, yaitu :

1. *Double* klik pada file *compile.bat* yang ada pada folder *ONE Simulator*,
2. Menjalankan dengan *command prompt* dengan masuk kedalam direktori *ONE Simulator* dan mengeksekusi program *compile.bat* seperti gambar 4.5.



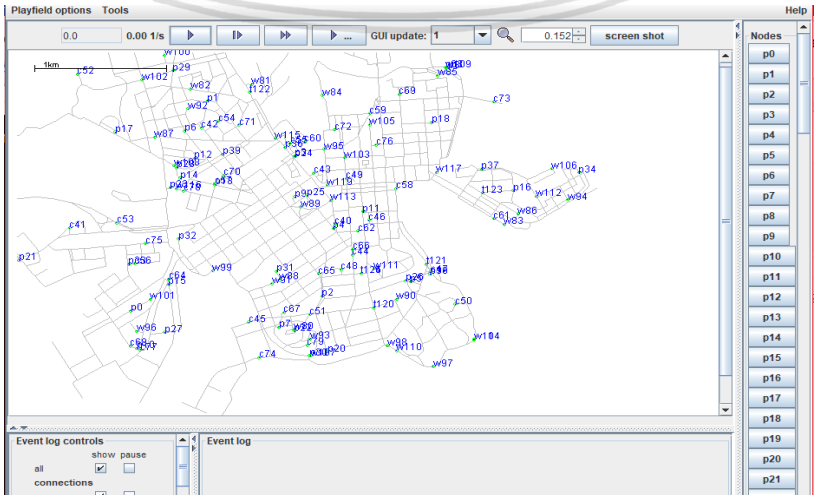
Gambar 4.5 Eksekusi *compile.bat* dengan *command prompt*

3. Setelah *compile* selesai dilakukan, *ONE Simulator* dapat dijalankan dengan cara mencari file *one.bat* dan melakukan *double* klik untuk mengeksekusi program seperti gambar 4.6.



Gambar 4.6 File *one.bat* pada folder *ONE Simulator*

4. Berikut adalah gambar tampilan *default* dari program *ONE Simulator* yang telah dijalankan dapat dilihat pada gambar 4.7.



Gambar 4.7 Tampilan *default* program *ONE Simulator.*

4.3 Skenario Pengujian

Skenario pengujian pada penelitian ini dibuat secara simulasi berdasarkan pengiriman data dengan rute jalan dari area kebun percobaan Cangar Universitas Brawijaya ke *data center* Universitas Brawijaya Malang. Simulasi dijalankan selama 12 jam dengan menggunakan *node* sebanyak 150 dengan kecepatan 10 – 50 Km/jam dan *Stationary Relay Node* sebanyak 10 serta dengan menggunakan *Buffer* sebesar 30Mb, 50Mb, 80Mb dan 100Mb. Simulasi dijalankan dengan algoritma *routing Multi Copy* yang telah ditentukan yakni *Epidemic, ProPHET* dan *Spray And Wait* dengan *management buffer First In – First Out* (FIFO).

Skenario pengujian pertama menggunakan *routing Epidemic* dengan *node* yang bergerak sebanyak 150 dan *Stationary Relay Node* sebanyak 10 dengan ukuran pesan sebesar 5Mb dan memberikan *buffer* sebesar 30Mb, 50Mb, 80Mb dan 100Mb serta menggunakan metode *Single Destination* dan *Multi Destination.* Skenario pengujian kedua menggunakan *routing ProPHET* dengan *node* yang bergerak sebanyak 150 dan *Stationary Relay Node* sebanyak 10 dengan ukuran pesan sebesar 5Mb dan memberikan *buffer* sebesar 30Mb, 50Mb, 80Mb dan 100Mb serta menggunakan *Single Destination* dan *Destination.* Skenario pengujian ketiga menggunakan *routing Spray And Wait* dengan *node* yang bergerak sebanyak 150 dan *Stationary Relay Node* sebanyak 10 dengan ukuran pesan sebesar 5Mb dan memberikan *buffer* sebesar 30Mb, 50Mb, 80Mb dan 100Mb serta menggunakan *Single Destination* dan *Multi Destination.* Tabel skenario simulasi pengujian dapat dilihat pada tabel 4.1.

Tabel 4.1 Skenario Simulasi Pengujian

No.	Skenario	Penjelasan	Source dan Destination	Buffer
1	Skenario 1	Simulasi menggunakan *routing Epidemic* dan Waktu simulasi selama 12 jam dengan *node* sebanyak 150 dengan kecepatan 10 – 50 Km/jam serta *Stationary Relay Node* sebanyak 10 dengan ukuran pesan sebesar 5Mb	*Single*	30Mb
				50Mb
			Multi	80Mb
				100Mb
2	Skenario 2	Simulasi menggunakan *routing ProPHET* dan Waktu simulasi selama 12 jam dengan *node* sebanyak 150 dengan kecepatan 10 – 50 Km/jam serta *Stationary Relay Node* sebanyak 10 dengan ukuran pesan sebesar 5Mb	*Single*	30Mb
				50Mb
			Multi	80Mb
				100Mb
3	Skenario 3	Simulasi menggunakan *routing Spray And Wait* dan Waktu simulasi selama 12 jam dengan *node*	*Single*	30Mb
				50Mb

		sebanyak 150 dengan kecepatan 10 – 50 Km/jam serta *Stationary Relay Node* sebanyak 10 dengan ukuran pesan sebesar 5Mb	*Multi*	80Mb
				100Mb

4.3.1 Konfigurasi Default Setting Skenario

Berikut ini merupakan hal yang perlu dikonfigurasi pada *default setting* untuk menjalankan skenario.

1. Konfigurasi lama waktu simulasi dijalankan untuk mengirimkan pesan pada skenario *ONE Simulator* adalah 12 jam dengan satuan detik seperti tabel 4.2 berikut ini :

Tabel 4.2 Kode sumber konfigurasi waktu skenario

```
Default_setting : konfigurasi waktu skenario
1  ## Scenario settings
2  Scenario.name = %%Group.router%%_%%Group.bufferSize%%_single
3  Scenario.simulateConnections = true
4  Scenario.updateInterval = 0.1
5  # 43200s == 12h
6  # 54000s == 15h
7  Scenario.endTime = 43200
8  btInterface.transmitSpeed = 250k
9  btInterface.transmitRange = 25
```

Keterangan :

- Baris 2 digunakan untuk memberi nama file hasil dari simulasi yang dijalankan.
- Baris 3 mendeklarasikan skenario koneksi bersifat *true.*
- Baris 4 menunjukan update data dalam simulator yang berjalan dengan interval waktu 1 detik.
- Baris 5 – 7 mendeklarasikan lama waktu skenario dijalankan pada simulator yaitu selama 43200 detik atau 12 jam.
- Baris 8 – 9 mendeklarasaikan kecepatan transfer data dari perangkat sebesar 2,5Mb per detik dan cakupan area sebesar 25 meter.

2. Konfigurasi jumlah *group node* dan *Stationary Relay Node* pada *ONE Simulator* untuk menjalankan skenario. Jumlah *group* pada skenario ini adalah 1 *group* dengan jumlah *node* 150 dan jumlah *Stationary Relay Node* sebanyak 10 *node* yang telah ditempatkan pada koordinat tertentu sepanjang jalur pengiriman pesan. Kode sumber dari konfigurasi *group node* dan *Stationary Relay Node* seperti pada tabel 4.3.

Tabel 4.3 Kode sumber konfigurasi *node* dan *Stationary Relay Node*

Default_setting : konfigurasi node dan Stationary Relay Node

```
# Define 11 different node groups
Scenario.nrofHostGroups = 11
# node yang bergerak sebanyak 150 bergerak dengan penambahan
# relaynode sebanyak 10 buah
# mobil dan angkot 10-50 km/h
Group.speed = 2.7, 13.9

Group.nrofHosts = 150
# group1  specific settings
Group1.groupID = node

#Group2 Relay node ATAS
Group2.movementModel = StationaryMovement
Group2.nrofHosts = 1
Group2.nodeLocation = 2102.0,38.0
Group2.groupID = Relay
Group2.speed = 0,0

#Group3 Relay node ATAS tengah
Group3.movementModel = StationaryMovement
Group3.nrofHosts = 1
Group3.nodeLocation = 1952.0,1690.0
Group3.groupID = Relay
Group3.speed = 0,0

#Group4 Relay node ATAS tengah
Group4.movementModel = StationaryMovement
Group4.nrofHosts = 1
Group4.nodeLocation = 370.0,3590.0
Group4.groupID = Relay
Group4.speed = 0,0

#Group5 Relay node ATAS tengah
Group5.movementModel = StationaryMovement
Group5.nrofHosts = 1
Group5.nodeLocation = 496.0,6378.0
Group5.groupID = Relay
Group5.speed = 0,0

```

```
#Group6 Relay node ATAS tengah
Group6.movementModel = StationaryMovement
Group6.nrofHosts = 1
Group6.nodeLocation = 4498.0,4887.0
Group6.groupID = Relay
Group6.speed = 0,0

#Group7 Relay node TEngah
Group7.movementModel = StationaryMovement
Group7.nrofHosts = 1
Group7.nodeLocation = 8840.0, 4545.0
Group7.groupID = Relay
Group7.speed = 0,0

#Group8 Relay node TEngah
Group8.movementModel = StationaryMovement
Group8.nrofHosts = 1
Group8.nodeLocation = 12140.0, 4992.0
Group8.groupID = Relay
Group8.speed = 0,0

#Group9 Relay node TEngah
Group9.movementModel = StationaryMovement
Group9.nrofHosts = 1
Group9.nodeLocation = 15680.0, 5229.0
Group9.groupID = Relay
Group9.speed = 0,0

#Group10 Relay node TEngah
Group10.movementModel = StationaryMovement
Group10.nrofHosts = 1
Group10.nodeLocation = 19246.0, 5672.0
Group10.groupID = Relay
Group10.speed = 0,0

#Group11 Relay node
Group11.movementModel = StationaryMovement
Group11.nrofHosts = 1
Group11.nodeLocation = 21926.0,6555.0
Group11.groupID = Relay
Group11.speed = 0,0
```

Keterangan :

- Baris 1 - 4 mendefinisikan *host* grup pada skenario yang berjumlah 11, yaitu 1 grup dengan *node* yang bergerak sebanyak 150 *node* dan 10 grup menjadi *Stationary Relay node.*
- Baris 5 – 10 mendeklarasikan grup *host* ke 1 dengan *node* sebanyak 150 *node* serta id grup bernama *node* dan bergerak pada route dengan kecepatan 10 sampai 50 km/jam.
- Baris 13 – 17 mendeklarasikan grup *host* ke 2 dengan jumlah *host* 1 *node* dan pemodelan *node* bergerak sebagai *Stationary relay node* yang diletakan pada lokasi tertentu yaitu koordinat 2102.0, 38.0 dengan id grup bernama *Relay.* Karena dijadikan sebagai *relay node* maka grup 2 memiliki kecepatan bergerak 0 km/jam.
- Baris 20 – 24 mendeklarasikan grup *host* ke 3 dengan jumlah *host* 1 *node* dan pemodelan *node* bergerak sebagai *Stationary relay node* yang diletakan pada lokasi tertentu yaitu koordinat 1952.0, 1690.0 dengan id grup bernama *Relay.* Karena dijadikan sebagai *relay node* maka grup 3 memiliki kecepatan bergerak 0 km/jam.
- Baris 27 – 31 mendeklarasikan grup *host* ke 4 dengan jumlah *host* 1 *node* dan pemodelan *node* bergerak sebagai *Stationary relay node* yang diletakan pada lokasi tertentu yaitu koordinat 370.0, 3590.0 dengan id grup bernama *Relay.* Karena dijadikan sebagai *relay node* maka grup 4 memiliki kecepatan bergerak 0 km/jam.
- Baris 34 – 38 mendeklarasi grup *host* 5 dan memiliki karakteristik yang sama dengan baris 27 – 31 tetapi *Stationary relay node* diletakan pada koordinat 496.0, 6378.0 dengan id grup bernama *Relay.*
- Baris 41 – 45 mendeklarasi grup *host* 6 dan memiliki karakteristik yang sama dengan baris 27 – 31 tetapi *Stationary relay node* diletakan pada koordinat 4498.0, 4887.0 dengan id grup bernama *Relay.*
- Baris 48 – 52 mendeklarasi grup *host* 7 dan memiliki karakteristik yang sama dengan baris 27 – 31 tetapi *Stationary relay node* diletakan pada koordinat 8840.0, 4545.0 dengan id grup bernama *Relay.*
- Baris 55 – 59 mendeklarasi grup *host* 8 dan memiliki karakteristik yang sama dengan baris 27 – 31 tetapi *Stationary relay node* diletakan pada koordinat 12140.0, 4992.0 dengan id grup bernama *Relay.*
- Baris 62 – 66 mendeklarasi grup *host* 9 dan memiliki karakteristik yang sama dengan baris 27 – 31 tetapi *Stationary relay node* diletakan pada koordinat 15680.0, 5229.0 dengan id grup bernama *Relay.*

- Baris 69 – 73 mendeklarasi grup *host* 10 dan memiliki karakteristik yang sama dengan baris 27 – 31 tetapi *Stationary relay node* diletakan pada koordinat 19246.0, 5672.0 dengan id grup bernama *Relay.*
- Baris 76 – 80 mendeklarasikan grup *host* ke 11 dengan jumlah *host* 1 *node* dan pemodelan *node* bergerak sebagai *Stationary relay node* yang diletakan pada lokasi tertentu yaitu koordinat 21926.0, 6555.0 dengan id grup bernama *Relay.* Karena dijadikan sebagai *relay node* maka grup 9 memiliki kecepatan bergerak 0 km/jam.

3. Konfigurasi *routing* dan ukuran *buffer* untuk menentukan *routing* apa dan berapa besar batasan *buffer* pada *node* untuk simulasi. Pada skenario ini *routing* yang digunakan adalah 3 *routing Multi Copy* yaitu *Epicemic, ProPHET,* dan *Spray And Wait.* Besar *buffer* yang digunakan juga memiliki 4 ukuran, yaitu sebesar 30Mb, 50Mb, 80Mb dan 100Mb. Kode sumber dari konfigurasi *routing* dan *buffer* yang digunakan dapat dilihat pada tabel 4.4.

Tabel 4.4 Kode sumber konfigurasi *Routing* dan ukuran *Buffer*

Default_setting : konfigurasi *Router group* dan *Buffer Size*	
1	>>>>>>>> Common settings for all groups
2	#skenario pergerakan node, protokol routing dan buffer
3	Group.movementModel = ShortestPathMapBasedMovement
4	Group.router = [EpidemicRouter; ProphetRouter; SprayAndWaitRouter;]
5	Group.bufferSize = [30M; 50M; 80M; 100M;]
6	#skenario TTl tiap pesan
7	# Message TTL of 420 minutes (7 hours)
8	Group.msgTtl = 420

Keterangan :

- Baris 3 mendeklarasikan pergerakan *node* pada jalur dengan model *Shortest Path Map Based Movent.*
- Baris 4 mendeklarasikan protokol *routing* yang dipakai yaitu *Epicemic, ProPHET,* dan *Spray And Wait.*
- Baris 5 mendeklarasikan ukuran *buffer* yang digunakan dalam menjalankan skenario yang telah dibuat yaitu sebesar 30Mb, 50Mb, 80Mb dan 100Mb.
- Baris 8 mendeklarasikan *Time To Live* (TTL) pesan dalam jaringan selama 420 menit atau 7 jam.

4. Konfigurasi ukuran pesan dan *Source and Destination* dari pesan yang dikirimkan sesuai dengan kebutuhan pada pelaksanaan skenario simulasi. Pada skenario pengujian ini besar ukuran dari pesan yang dikirimkan telah ditetapkan sebesar 5Mb dengan interval persoupesan yang dibuat 35 detik. *Source and Destination* pada pengujian skenario ini dibuat menjadi *Single Destination* dan *Multi Destination*. Pada *Single Destination* alamat dari

pengirim dan penerima pesan hanya dibuat 1 alamat, sedangkan *Multi Destination* alamat pengirim dari satu sumber yang sama dan alamat tujuan akan dibuat menjadi 2 alamat. Pada konfigurasi *multi destination* maka kode sumber akan ditambahkan 1 *event generator* dan apa bila konfigurasi menggunakan *single destination* maka *even generator* hanya ada 1. Kode sumber dari konfigurasi ukuran pesan dan *Source and Destination* dapat dilihat pada tabel 4.5.

Tabel 4.5 Konfigutasi ukuran pesan dan *Source and Destination*

Default_setting : konfigurasi ukuran pesan dan *Source and Destination*	
1	## Message creation parameters
2	# How many event generators
3	Events.nrof = 2
4	# Class of the first event generator
5	Events1.class = MessageEventGenerator
6	# Creation interval in seconds (one new message every 35 seconds)
7	Events1.interval = 35
8	Events1.size = 5M
9	# range of message source/destination addresses
10	Events1.hosts = 0,155
11	# Message ID prefix
12	Events1.prefix = Single
13	# Class of the second event generator
14	Events2.class = MessageEventGenerator
15	# Creation interval in seconds (one new message every 35 seconds)
16	Events2.interval = 35
17	Events2.size = 5M
18	# range of message source/destination addresses
19	Events2.hosts = 0, 156
20	# Message ID prefix
21	Events2.prefix = Multipel

Keterangan :

- Baris 1 – 3 mendeklarasikan jumlah *event generator* pada skenario sebanyak 2 *event.*
- Baris 4 – 5 mendeklarasikan *event class* pertama adalah *Message Event Generator* yaitu mempublikasikan pesan sebagai pengiriman dari sumber ke tujuan.
- Baris 6 – 8 merupakan pembuatan waktu untuk *event* 1 dengan interval waktu 35 detik untuk pembuatan 1 pesan pada node sumber dengan ukuran pesan sebesar 5Mb.

- Baris 9 – 12 range dari sumber pesan dan *node* tujuan yaitu dengan *node* 1 tetapi dimulai dari 0 karena menggunakan tipe data *array, node* 1 sebagai sumber dan *node* 155 sebagai tujuan pesan. Pada *event generator* ke 1 setiap paket data memiliki id *prefix Single*.
- Baris 13 – 14 mendeklarasikan *event class* yang ke 2 adalah *Message Event Generator* yaitu mempublikasikan pesan sebagai pengiriman dari sumber ke tujuan. *event class* yang ke 2 ini digunakan untuk skenario *Multiple Destination.*
- Baris 15 – 17 merupakan pembuatan waktu untuk *event* ke 2 dengan interval waktu 35 detik untuk pembuatan 1 pesan pada node sumber dengan ukuran pesan sebesar 5Mb.
- Baris 18 – 21 range dari sumber pesan dan *node* tujuan yaitu dengan *node* 1 tetapi dimulai dari 0 karena menggunakan tipe data *array, node* 1 sebagai sumber dan *node* 156 sebagai tujuan pesan. Pada *event generator* ke 2 setiap paket data memiliki id *prefix Single*.

5. Konfigurasi *Management Buffer First In - First Out* (FIFO) agar paket yang masuk ke dalam *buffer node* dapat beraturan sehingga apabila *buffer node* penuh dan ada paket baru yang akan diakomodasi oleh *node* tersebut maka pesan yang telah lama berada pada *buffer* akan di *drop* untuk memberikan ruang sehingga pesan baru dapat diakomodasi oleh *node*. Penghapusan pesan pada *buffer node* dilakukan sesuai urutan waktu saat pesan masuk pada *node*. Pesan pada *buffer node* akan dihapus sesui antrian sampai *buffer* cukup untuk mengakomodasi pesan baru. Kode sumber *Pseudocode* pada Konfigurasi *Management Buffer First In - First Out* (FIFO) dapat dilihat pada tabel 4.6

Tabel 4.6 Kode sumber Pseudocode *Management Buffer* (FIFO)

Speudocode FIFO : konfigurasi *Management Buffer* FIFO	
1	`Void write (fifo, val)`
2	`{`
3	`  while (ruang_tersedia = 0)`
4	`  ruang_tersedia = get_ruang (fifo);`
5	`  Store (wptr++, val);`
6	`  If (wptr == buffer_akhir)`
7	`  wptr = buffer_awal;`
8	`  ruang_tersedia-;`
9	`  data_produced++;`
10	`  If (data_produced == data_produced_limit)`
11	`  {`
12	`  ambil_data (fifo, data_produced);`
13	`  data_produced = 0;`
14	`  }`

Sumber : (Bhattacarya, et al., 2003)

Keterangan :

- Baris 1 deklarasi penulisan metode fifo
- Baris 3 – 5 apabila rungan *buffer* = 0 atau penuh, maka variabel ruang_tersedia akan meminta ruang *buffer* kepada kontroler tetapi bila kontorler merespon bahwa tidak ada *buffer* kosong maka akan tetap meminta sampai get_ruang merespon ruang_tersedia tidak kosong.
- Baris 6 – 7 jika wptr merupakan data yang tersimpan didalam *buffer* merupakan data terakhir atau berada pada *buffer* ekor maka wptr sama dengan data dalam *buffer* diawal yang akan di hapus.
- Baris 8 – 9 variabel ruang_tersedia akan berkurang dengan masuknya data dalam *buffer* dan variabel data_produced akan di tambah satu sebagaimana data yang ditambahkan pada ruang yang tersedia dalam *buffer.*
- Baris 10 – 13 mendeklarasokan jika variabel data_produce adalah variabel data_produce_limit dan apabila limit dalam *buffer* sudah penuh tetapi ada permintaan kesediaan ruang pada *buffer* maka data akan diambil atau dihapus dengan prosedur fifo dalam data pada variabel data_produced.

4.4 Pengujian Skenario

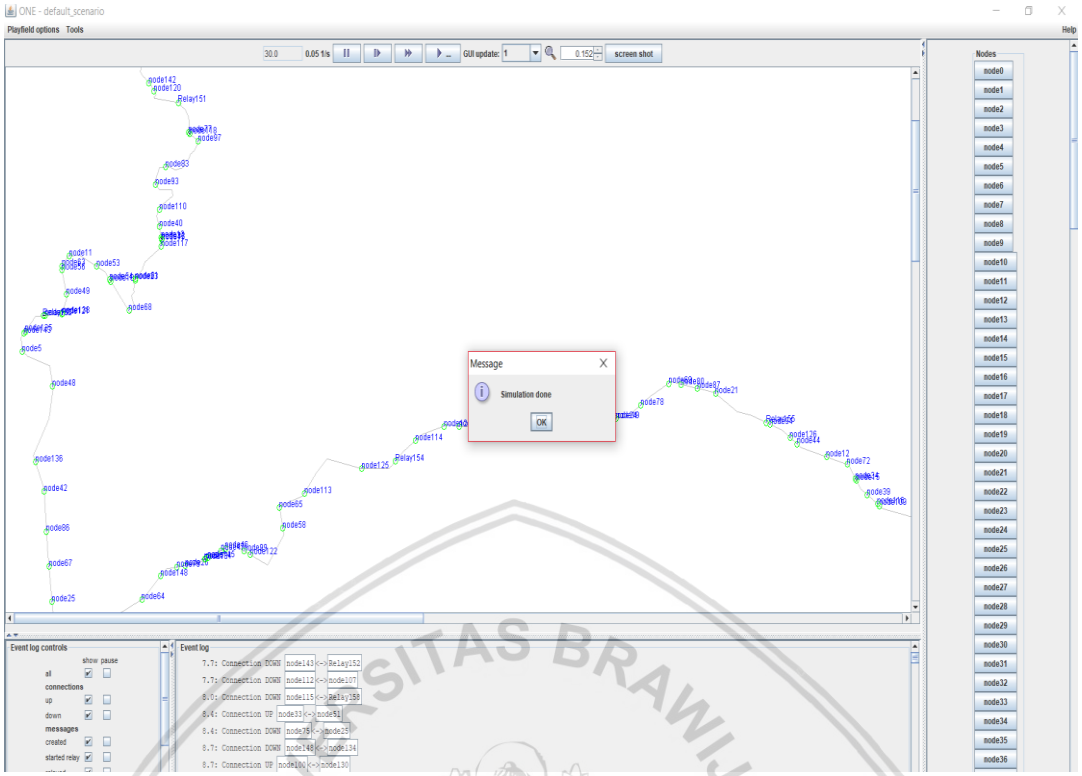
Pengujian skenario dilakukan dengan 6 tahap sesuai dengan skenario yang telah dibuat sebelumnya. Skenario akan dijalankan menggunakan aplikasi *The ONE Simulator*.

4.4.1 Pengujian Skenario Satu

Pada pengujian Skenario satu dilakukan untuk mengetahui kinerja dari *routing Epidemic* dengan ukuran *buffer* sebesar 30Mb, 50Mb, 80Mb dan 100Mb serta *Single Destination.* Pada tabel 4.7 menunjukan skenario pengujian satu dan hasil simulasi dari pengujian *ONE Simulator* di tunjukan pada gambar 4.8.

Tabel 4.7 Skenario Pengujian Satu

Skenario	Penjelasan	*Source and Destination*	Ukuran Buffer
Skenario 1	Simulasi dengan menggunakan *routing* protokol *Epidemic* dengan *node* sebanyak 150 dan *Stationary Relay Node* sebanyak 10 serta *management buffer* FIFO	*Single*	30Mb
			50Mb
			80Mb
			100Mb



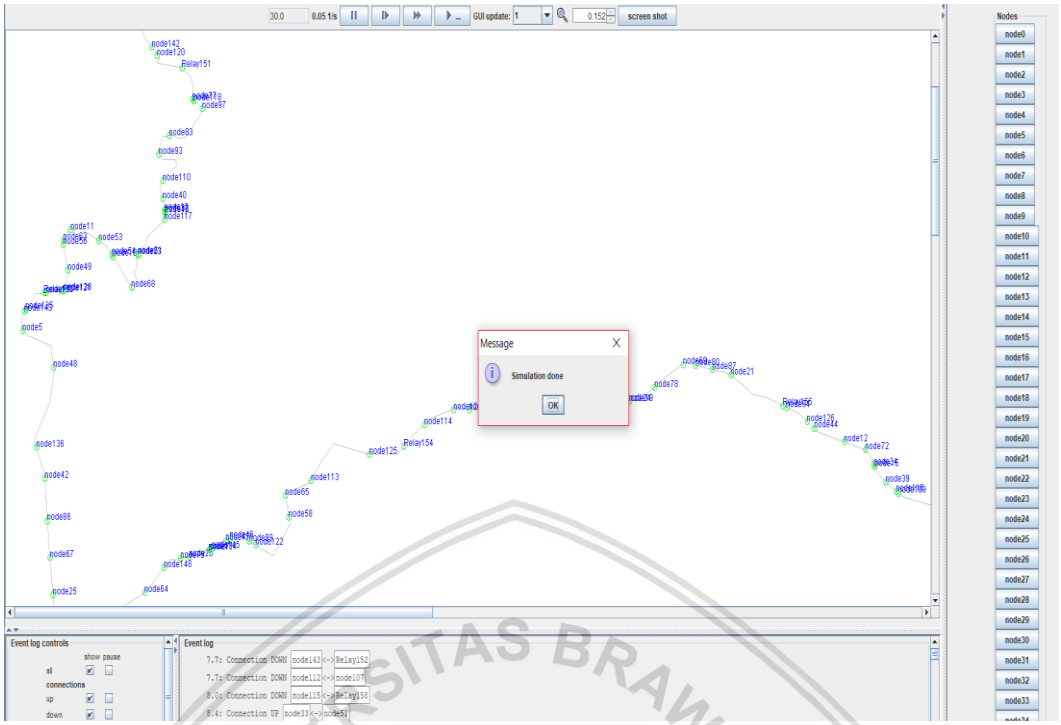
Gambar 4.8 Hasil dari pengujian satu pada aplikasi *ONE Simulator*

4.4.2 Pengujian Skenario Dua

Pada pengujian Skenario dua dilakukan untuk mengetahui kinerja dari *routing Epidemic* dengan ukuran *buffer* sebesar 30Mb, 50Mb dan 80Mb serta *Single Multi Destination.* Pada tabel 4.8 menunjukan skenario pengujian dua dan hasil simulasi dari pengujian *ONE Simulator* ditunjukan pada gambar 4.9.

Tabel 4.8 Skenario Pengujian Satu

Skenario	Penjelasan	*Source and Destination*	Ukuran Buffer
Skenario 2	Simulasi dengan menggunakan *routing* protokol *Epidemic* dengan *node* sebanyak 150 dan *Stationary Relay Node* sebanyak 10 *management buffer* FIFO	*Multi Destination*	30Mb
			50Mb
			80Mb
			100Mb



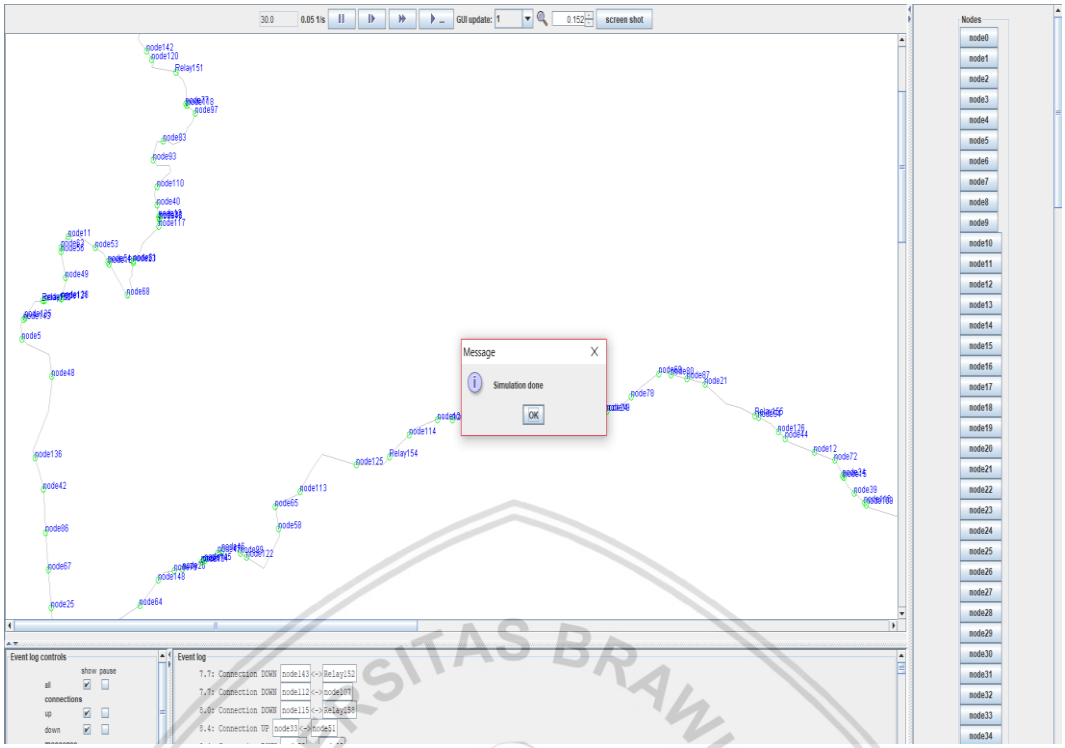
Gambar 4.9 Hasil dari pengujian dua pada aplikasi *ONE Simulator*

4.4.3 Pengujian Skenario Tiga

Pada pengujian Skenario tiga dilakukan untuk mengetahui kinerja dari *routing ProPHET* dengan ukuran *buffer* sebesar 30Mb, 50Mb dan 80Mb serta *Single Destination.* Pada tabel 4.9 menunjukan skenario pengujian tiga dan hasil simulasi dari pengujian *ONE Simulator* di tunjukan pada gambar 4.10.

Tabel 4.9 Skenario Pengujian Satu

Skenario	Penjelasan	*Source and Destination*	Ukuran Buffer
Skenario 3	Simulasi dengan menggunakan *routing* protokol *ProPHET* dengan *node* sebanyak 150 dan *Stationary Relay Node* sebanyak 10 *management buffer* FIFO	*Single*	30Mb
			50Mb
			80Mb
			100Mb



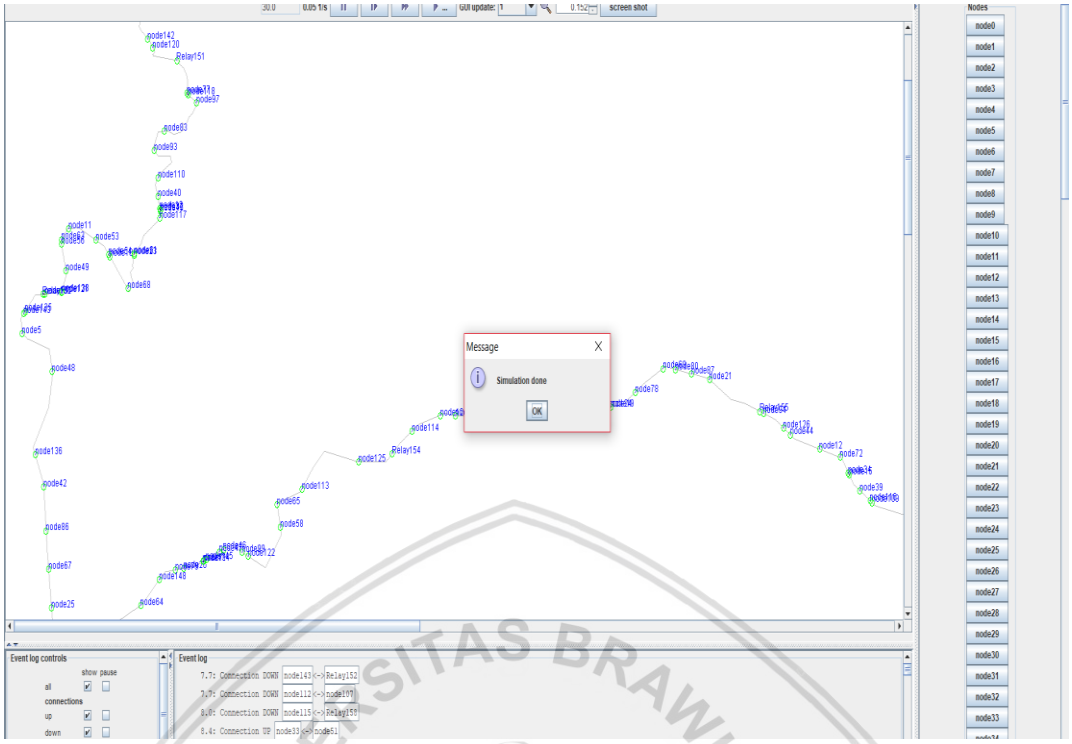
Gambar 4.10 Hasil dari pengujian tiga pada aplikasi *ONE Simulator*

4.4.4 Pengujian Skenario Empat

Pada pengujian Skenario empat dilakukan untuk mengetahui kinerja dari *routing ProPHET* dengan ukuran *buffer* sebesar 30Mb, 50Mb dan 80Mb serta *Single Multi Destination.* Pada tabel 4.10 menunjukan skenario pengujian empat dan hasil simulasi pengujian pada *ONE Simulator* ditunjukan pada gambar 4.11.

Tabel 4.10 Skenario Pengujian Satu

Skenario	Penjelasan	*Source and Destination*	Ukuran Buffer
Skenario 4	Simulasi dengan menggunakan *routing* protokol *ProPHET* dengan *node* sebanyak 150 dan *Stationary Relay Node* sebanyak 10 serta*management buffer* FIFO	*Multi Destination*	30Mb
			50Mb
			80Mb
			100Mb



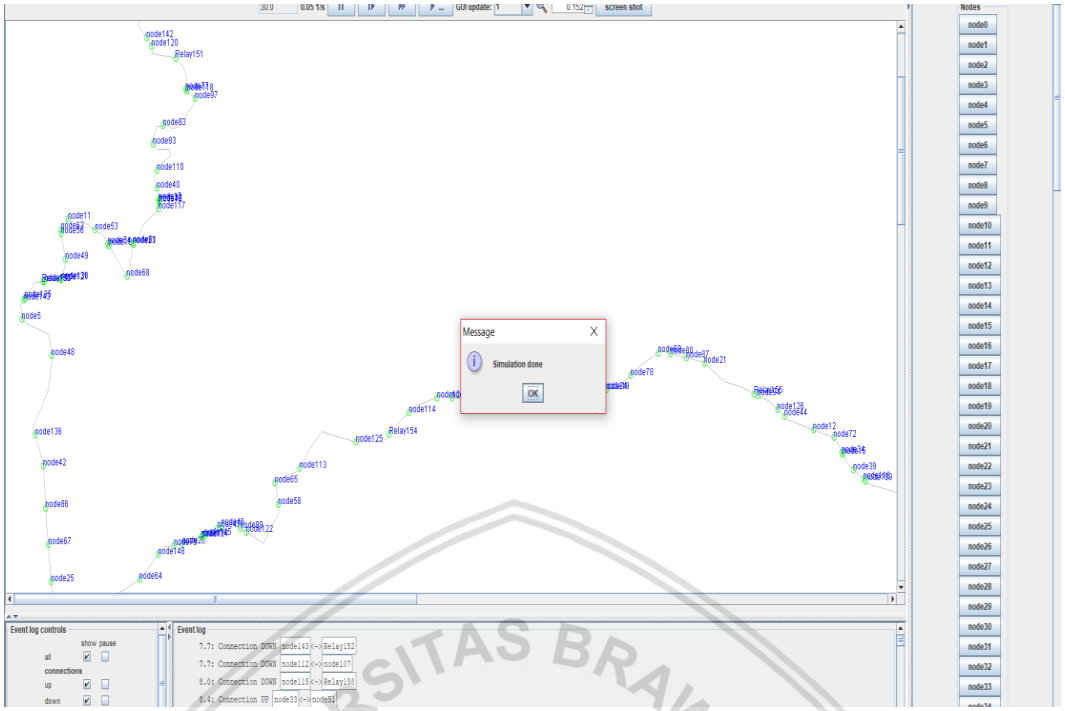
Gambar 4.11 Hasil dari pengujian empat pada aplikasi *ONE Simulator*

4.4.5 Pengujian Skenario Lima

Pada pengujian Skenario lima dilakukan untuk mengetahui kinerja dari *routing Spray And Wait* dengan ukuran *buffer* sebesar 30Mb, 50Mb dan 80Mb serta *Single Destination.* Pada tabel 4.11 menunjukan skenario pengujian lima dan hasil simulasi dari pengujian *ONE Simulator* di tunjukan pada gambar 4.12.

Tabel 4.11 Skenario Pengujian Satu

Skenario	Penjelasan	*Source and Destination*	Ukuran Buffer
Skenario 5	Simulasi dengan menggunakan *routing* protokol *Spray And Wait* dengan *node* sebanyak 150 dan *Stationary Relay Node* sebanyak 10 *management buffer* FIFO	*Single*	30Mb
			50Mb
			80Mb
			100Mb



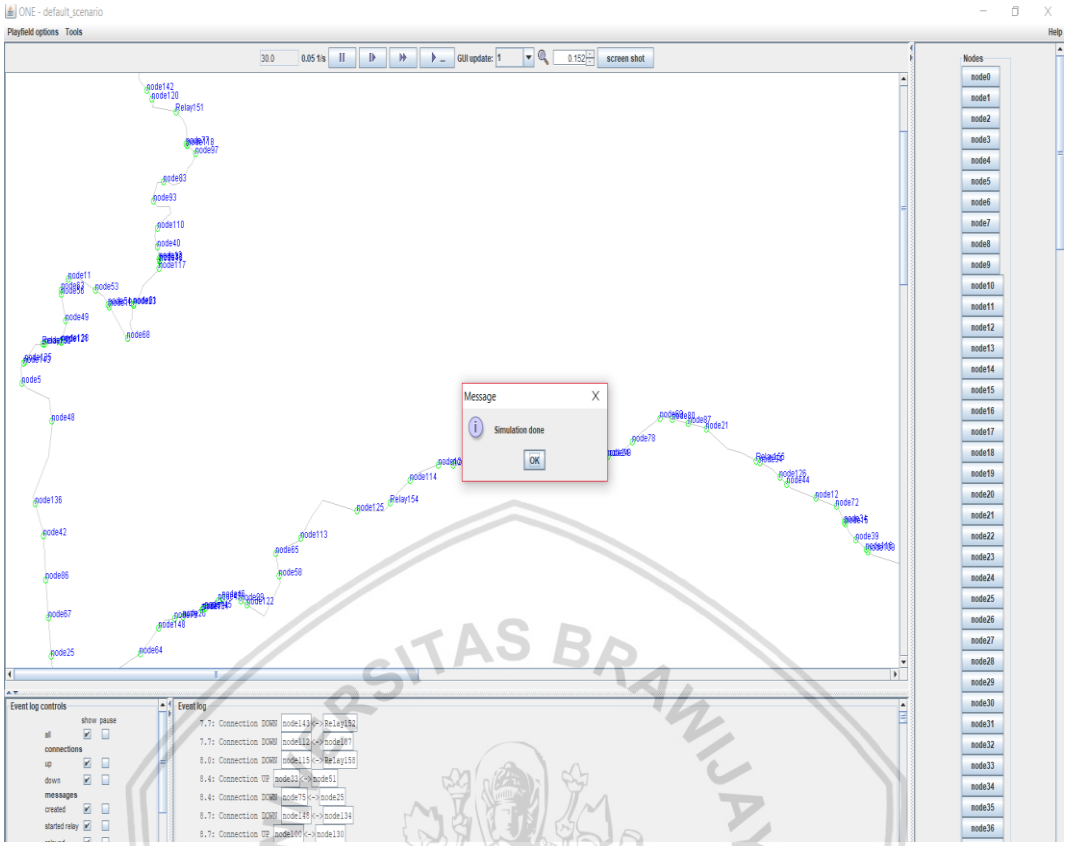
Gambar 4.12 Hasil dari pengujian lima pada aplikasi *ONE Simulator*

4.4.6 Pengujian Skenario Enam

Pada pengujian Skenario enam dilakukan untuk mengetahui kinerja dari *routing Spray And Wait* dengan ukuran *buffer* sebesar 30Mb, 50Mb dan 80Mb serta *Single Multi Destination.* Pada tabel 4.12 menunjukan skenario pengujian enam dan hasil simulasi dari pengujian pada *ONE Simulator* ditunjukan pada gambar 4.13.

Tabel 4.12 Skenario Pengujian Satu

Skenario	Penjelasan	*Source and Destination*	Ukuran Buffer
Skenario 6	Simulasi dengan menggunakan *routing* protokol *Spray And Wait* dengan *node* sebanyak 150 dan *Stationary Relay Node* sebanyak 10 *management buffer* FIFO	*Multi Destination*	30Mb
			50Mb
			80Mb
			100Mb



Gambar 4.13 Hasil dari pengujian enam pada aplikasi *ONE Simulator*

BAB 5 HASIL DAN ANALISIS

Setelah mendapatkan nilai hasil dari implementasi skenario yang telah di buat sesuai dengan skenario pada bab 4. Selanjutnya akan dilakukan analisis kinerja dari protokol *routing Multi copy* dengan penambahan *Stationary Relay Node* dan *management buffer* dengan algoritma *First in - First Out*, maka pada bab ini akan dicari nilai *Delivery Probability* dan *Overhead Ratio*nya.

5.1 *Delivery Probability*

Delivery Probability adalah perbandingan antara rasio dari banyaknya jumlah total pesan yang dibuat dan banyaknya pesan yang sampai ke tujuan. *Delivery Probability* menunjukan tingkat keberhasilah sebuah protokol *routing* dalam mendistribusikan pesan ke tujuan. Apabila nilai dari *Delivery Probability* semakin tinggi, maka pesan yang sampai ke tujuan akan semakin banyak. Berikut adalah tabel nilai *Delivery Probability* hasil pengujian dari skenario yang telah dibuat ditunjukan pada tabel 5.1 dan 5.2.

Tabel 5.1 Hasil nilai *Delivery Probability* pada skenario *Single Destination*

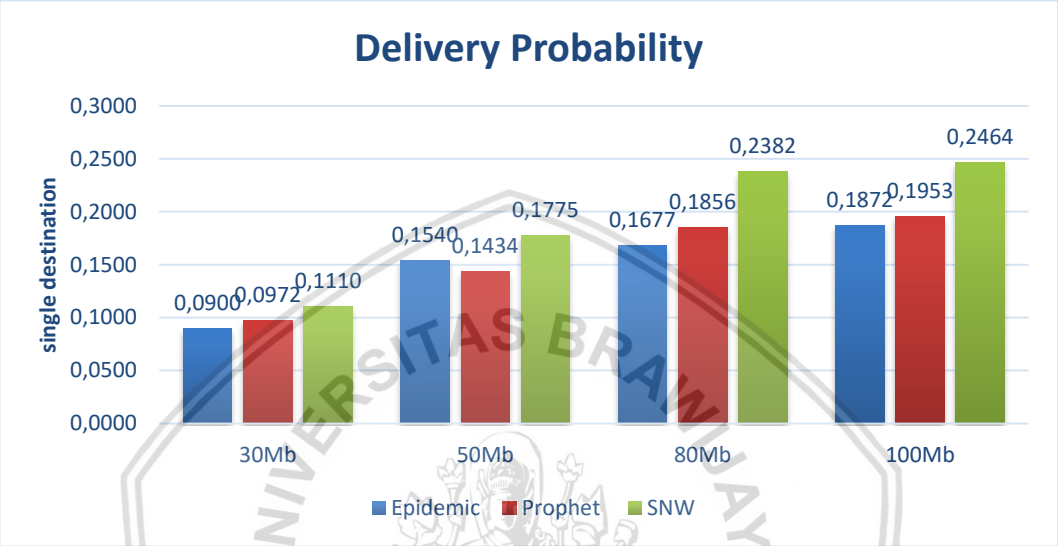
Delivery Probability			
Buffer	Epidemic	ProPHET	SNW
30Mb	0,0900	0,0972	0,1110
50Mb	0,1540	0,1434	0,1775
80Mb	0,1677	0,1856	0,2382
100Mb	0,1872	0,1953	0,2464

Pada tabel 5.1 menunjukan hasil nilai *Delivery Probability* dari pengujian skenario 1, 3 dan 5 dengan mekanisme skenario *Single Destination.* Semua skenario pengujian dijalankan selama 12 jam waktu simulator dengan *node* bergerak sebanyak 150 dengan kecepatan 10 – 50 Km/jam, *Stationary Relay Node* sebanyak 10 serta *Management Buffer* FIFO dengan ukuran pesan sebesar 5Mb dan ukuran *buffer* sebesar 30Mb, 50Mb, 80Mb dan 100Mb.

Tabel 5.2 Hasil nilai *Delivery Probability* pada skenario *Multi Destination*

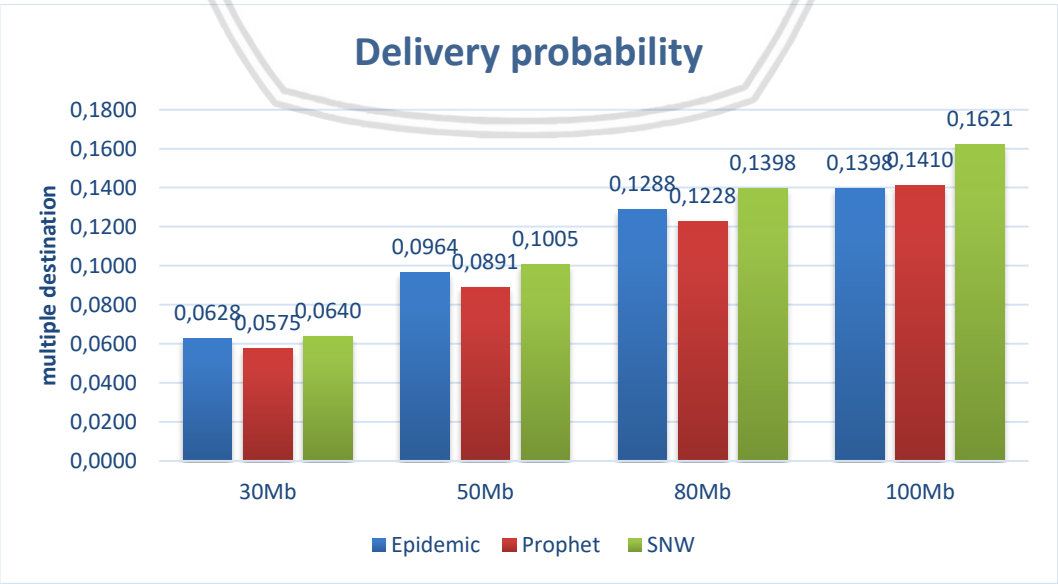
Delivery Probability			
Buffer	Epidemic	ProPHET	SNW
30Mb	0,0628	0,0575	0,0640
50Mb	0,0964	0,0891	0,1005
80Mb	0,1288	0,1228	0,1398
100Mb	0,1398	0,1410	0,1621

Pada tabel 5.2 menunjukan hasil nilai *Delivery Probability* dari pengujian skenario 2, 4 dan 6 dengan mekanisme skenario *Multiple Destination.* Semua skenario pengujian dijalankan selama 12 jam waktu simulator dengan *node* bergerak sebanyak 150 dengan kecepatan 10 – 50 Km/jam, *Stationary Relay Node* sebanyak 10 serta *Management Buffer* FIFO dengan ukuran pesan sebesar 5Mb dan ukuran *buffer* sebesar 30Mb, 50Mb, 80Mb dan 100Mb.



Gambar 5.1 Grafik *Delivery Probability* pada skenario *Single Destination*

Pada grafik 5.1 diatas merupakan bentuk data diagram yang dibuat berdasarkan tabel 5.1 dari pengujian skenario menggunakan ukuran *buffer* 30Mb, 50Mb, 80Mb dan 100Mb dengan protokol *routing Epidemic, ProPHET* dan *Spray And Wait* terhadap parameter uji *Delivery Probability* yang didapat berdasarkan skenario pengiriman pesan dengan mekanisme *Single Destination.*



Gambar 5.2 Grafik *Delivery Probability* pada skenario *Multi Destination*

Pada grafik 5.2 diatas merupakan bentuk data diagram yang dibuat berdasarkan tabel 5.2 dari pengujian skenario menggunakan ukuran *buffer* 30Mb, 50Mb, 80Mb dan 100Mb dengan protokol *routing Epidemic, ProPHET* dan *Spray And Wait* terhadap parameter uji *Delivery Probability* yang didapat bedasarkan skenario pengiriman pesan dengan mekanisme *Multiple Destination.*

Berikut ini merupakan penjelasan dan analisis dari grafik 5.1 dan 5.2 berdasarkan protokol *routing Epidemic, ProPHET* dan *Spray And Wait* pada skenario pengiriman pesan dengan mekanisme *single destination* dan *multiple destianion*.

1. *Routing Epidemic*

   Pada grafik gambar 5.1 dan gambar 5.2 menunjukan *routing Epidemic* terjadi kenaikan *delivery probability* pada setiap ukuran *buffer* pada skenario pengiriman *single destination* maupun *multiple destianion.* Pada ukuran *buffer* 30Mb *routing epidemic* memiliki kenaikan sebesar 9,0% dan 6,3% dengan nilai *delivery probability* sebesar 0,0900 dan 0,0628. Kemudian pada ukuran *buffer* 50Mb *routing epidemic* memiliki kenaikan sebesar 15,4% dan 9,6% dengan nilai *delivery probability* sebesar 0,1540 dan 0,0964. Selanjutnya pada ukuran *buffer* 80Mb *routing epidemic* memiliki kenaikan sebesar 16,7% dan 12,9% dengan nilai *delivery probability* sebesar 0,1677 dan 0,1288. Yang terakhir pada ukuran *buffer* 100Mb *routing epidemic* memiliki kenaikan sebesar 18,7% dan 13,9% dengan nilai *delivery probability* sebesar 0,1872 dan 0,1398.

   Pada *routing Epidemic* setiap pengujian skenario memiliki kenaikan nilai *delivery probability* hal ini terjadi karena pada *routing Epidemic* pesan akan terus disebarkan pada *node* di jaringan, sehingga setiap node akan memiliki salinan pesan. Pada *buffer* yang terbatas maka salinan pesan yang dapat dibawa oleh *node* memiliki keterbatasan pula, sehingga pada ukuran *buffer* yang lebih tinggi maka nilai *Delivery Probability*nya akan meningkat akan tetapi beban jaringan akan bertambah juga. Selain itu penambahan mekanisme *Stationary Relay Node* dapat menaikan nilai *Delivery Probability* karena memberikan jaminan untuk setiap *node* dapat saling bertemu, akan tetapi pada protokol *routing Epidemic* banyak sedikitnya *Stationary Relay Node* yang digunakan tidak terlalu dapat menaikan nilai *Delivery Probability* secara signigikan dikarenakan semua *node* pada jaringan pasti memiliki salinan pesan sebab penyebaran pesan pada *routing Epidemic* terjadi secara terus menerus dari *node* satu ke *node* yang lain.

   Dapat diamati perbedaan nilai *Delivery Probability* pada skenario *Single Destination* dan *Multiple Destination,* bahwa pada *Single Destination* nilai *Delivery Probability*nya lebih tinggi dari pada *Multiple Destination* yang terjadi pada semua ukuran *buffer*. Hal ini terjadi karena pada skenario *Multiple Destination* pesan yang dibuat untuk dikirimkan sejumalah 2468 jumlahnya dua kali lipat lebih banyak dari pada pesan yang dibuat pada skenario *Single Destination* yang hanya 1234 tetapi dengan batasan *buffer* yang sama.

Dengan mekanisme penyebaran pesan yang terjadi pada *Epidemic,* pesan yang dibawa oleh *node* akan lebih sering dihapus karena seringnya pesan baru yang datang dan disebarkan dari *node* lain secara terus menerus sedangkan *buffer* yang terbatas hanya dapat menampung jumlah pesan yang terbatas pula. Hal ini yang menjadikan *multiple destination* lebih sedikit pada rasio penyampain pesannya dari jumlah pesan yang dibuat pada *node* sumber dari pada *single destination* yang akhirnya berpengaruh pada nilai *delivery probability*nya. Sehingga dapat diambil kesimpulan bahwa *routing Epidemic* lebih efektif untuk mendistribusikan pesan dengan *buffer* yang sangat besar.

2. *Routing ProPHET*

Pada grafik gambar 5.1 dan gambar 5.2 menunjukan *routing ProPHET* terjadi juga kenaikan *delivery probability* pada setiap ukuran *buffer* pada skenario pengiriman *single destination* maupun *multiple destianion.* pada ukuran *buffer* 30Mb *routing ProPHET* memiliki kenaikan sebesar 9,7% dan 5,8% dengan nilai *delivery probability* sebesar 0,0972 dan 0,0575. Kemudian pada ukuran *buffer* 50Mb *routing ProPHET* memiliki kenaikan sebesar 14,3% dan 8,9% dengan nilai *delivery probability* sebesar 0,1434 dan 0,0891. Selanjutnya pada ukuran *buffer* 80Mb *routing ProPHET* memiliki kenaikan sebesar 18,6% dan 12,3% dengan nilai *delivery probability* sebesar 0,1856 dan 0,1228. Yang terakhir pada ukuran *buffer* 100Mb *routing ProPHET* memiliki kenaikan sebesar 19,5% dan 14,1% dengan nilai *delivery probability* sebesar 0,1953 dan 0,1410.

Pada skenario *Multiple Destination* dengan ukuran *buffer* 30Mb, 50Mb dan 80Mb *routing ProPHET* memiliki nilai yang lebih rendah dari *routing Epidemic* dan *Spray And Wait* hal ini terjadi karena pada *routing ProPHET* memiliki data *delivery predictability* untuk memprediksi *node* yang dapat menyampaikan pesan ke tujuan, dan dari data *delivery predictability* jumlah salinan akan diutamakan ke *node* yang memiliki nilai penyampaian tertinggi sehingga jumlah salinan pesan pada *ProPHET* memiliki jumlah yang terbatas. Selain itu ada hal yang dapat mempengaruhi rendahnya nilai *Delivery Probability* pada *routing ProPHET* yaitu karena banyaknya *node* yang menua sehingga tidak semua *node* dapat digunakan untuk menyampaikan pesan.

Node dapat menua karena tingkat *delivery predictability*nya menurun, hal itu disebabkan karena pertemuan antar *node* sudah jarang atau tidak bertemu lagi, tetapi pada *buffer* 100Mb nilai *delivery probability*nya lebih tinggi dari *routing Epidemic,* hal ini karena penambahan penambahan ukuran *buffer* dan mekanisme *Stationary Relay Node* dapat menaikan nilai *Delivery Probability* karena memberikan jaminan untuk setiap *node* dapat saling bertemu sehingga pada *routing ProPHET* tingkat *delivery predictability*nya dapat meningkat oleh karena itu salinan pesan ke *node* lain dapat terdistribusikan ke *node* yang mempunyai penyampaian pesan yang baik. Selain itu besarnya ukuran *buffer* dapat meningkatkan nilai *Delivery Probability* karena salinan pesan yang dapat dibawa *node* ke tujuan juga akan semakin banyak.

Dapat diamati perbedaan nilai *Delivery Probability* pada skenario *Single Destination* dan *Multiple Destination,* bahwa pada *Single Destination* nilai *Delivery Probability*nya lebih tinggi dari pada *Multiple Destination* pada semua ukuran *buffer*. Hal ini terjadi karena pada skenario *Multiple Destination* pesan yang dibuat untuk dikirimkan sejumalah 2468 jumlahnya dua kali lipat lebih banyak dari pada pesan yang dibuat pada skenario *Single Destination* yang hanya 1234 tetapi dengan batasan *buffer* yang sama. Dengan mekanisme penyebaran pesan yang terjadi pada *ProPHET,* keterbatasan *buffer* mempengaruhi penyampaian pesan ketujuan . Pada penyebaran pesan pada *ProPHET* memprioritaskan salinan pesan ke *node* yang mempunyai tingkat *delivery predictability* tinggi sehingga *node* yang mempunyai salinan pesan menjadi sedikit dan banyaknya pesan yang dibuat pada sumber membuat salinan pesan yang dibawa oleh *node* akan lebih sering dihapus karena seringnya pesan baru yang datang, sedangkan *buffer* yang terbatas hanya dapat menampung jumlah pesan yang terbatas pula. Hal ini yang menjadikan *multiple destination* lebih sedikit pada rasio penyampain pesannya dari jumlah pesan yang dibuat pada *node* sumber dari pada *single destination* yang akhirnya berpengaruh pada nilai *delivery probability*nya. Sehingga dapat diambil kesimpulan bahwa pada *routing Epidemic* skenario skenario *Single Destination* lebih efektif untuk mendistribusikan pesan dari pada skenario *Multiple Destination* meskipun dengan *buffer* yang besar.

3. *Routing Spray And Wait*

Pada grafik gambar 5.1 dan gambar 5.2 menunjukan *routing Spray And Wait* terjadi kenaikan *delivery probability* pada setiap ukuran *buffer* pada skenario pengiriman *single destination* maupun *multiple destianion.* pada ukuran *buffer* 30Mb *routing Spray And Wait* memiliki kenaikan sebesar 11,1% dan 6,4% dengan nilai *delivery probability* sebesar 0,1110 dan 0,0640. Kemudian pada ukuran *buffer* 50Mb *routing Spray And Wait* memiliki kenaikan sebesar 17,8% dan 10,1% dengan nilai *delivery probability* sebesar 0,1775 dan 0,1005. Selanjutnya pada ukuran *buffer* 80Mb *routing Spray And Wait* memiliki kenaikan sebesar 23,8% dan 14,0% dengan nilai *delivery probability* sebesar 0,2382 dan 0,1398. Yang terakhir pada ukuran *buffer* 100Mb *routing Spray And Wait* memiliki kenaikan sebesar 24,6% dan 16,2% dengan nilai *delivery probability* sebesar 0,2464 dan 0,1621.

Pada pengujian ini *routing Spray and Wait* memiliki nilai *delivery probability* tertinggi dari *routing Epidemic* maupun *routing ProPHET* baik dari skenario dengan *single destination* maupun *multiple destination,* hal ini terjadi karena *routing Spray and Wait* memiliki dua tahap dalam pengiriman. Tahap pertama ialah tahap *Spray,* pada tahap ini *routing Spray and Wait* melakukan penyebaran pesan ke semua *node* seperti pada *routing Epidemic* tetapi memiliki batasan penyebaran dengan jumlah tertentu, pada pengujian ini batas penyebaran sebanyak 6 *copy*. Selanjutnya apabila pada tahap *spray node* tujuan tidak ditemukan dan jumlah penyebaran telah mencapai batasan,

maka tahap ke dua yaitu tahap *wait* akan berjalan yakni pesan akan dibawa sampai bertemu dengan *node* tujuan.

Selain mekanisme penyebaran pesan pada *Spray and Wait,* penambahan ukuran *buffer* juga dapat menambah nilai *Delivery Probability* karena pesan yang disebarkan maupun yang disimpan dalam *buffer* akan semakin banyak, sehingga pesan dapat tersampaikan lebih banyak lagi saat *node* pembawa pesan sampai ke *node* tujuan. Pemakaian mekanisme *Stationary Relay Node* juga merupakan salah saru faktor yang dapat menaikan nilai *Delivery Probability* karena memberikan jaminan untuk setiap *node* dapat saling bertemu sehingga pada tahap *Spray* salinan pesan ke *node* lain akan semakin banyak.

Dapat diamati perbedaan nilai *Delivery Probability* pada *Spray and Wait* dengan skenario *Single Destination* dan *Multiple Destination,* bahwa pada *Single Destination* nilai *Delivery Probability*nya lebih tinggi dari pada *Multiple Destination* pada semua ukuran *buffer* yang diujikan. Hal ini terjadi karena pada skenario *Multiple Destination* pesan yang dibuat untuk dikirimkan sejumalah 2468 jumlahnya dua kali lipat lebih banyak dari pada pesan yang dibuat pada skenario *Single Destination* yang hanya 1234 tetapi dengan batasan *buffer* yang sama. Dengan mekanisme penyebaran pesan yang terjadi pada *Spray and Wait,* pada tahap *Spray* pesan yang dibawa oleh *node* akan lebih sering dihapus karena seringnya pesan baru yang datang dan disebarkan dari *node* lain secara terus menerus sedangkan *buffer* yang terbatas hanya dapat menampung jumlah pesan yang terbatas pula. Banyaknya pesan yang dibuat juga berpengaruh pada fase *Wait,* banyak salinan pesan terduplikasi pada *node* satu dengan *node* yang lain. Sehingga menjadikan *multiple destination* lebih sedikit pada rasio penyampain pesannya dari jumlah pesan yang dibuat pada *node* sumber dari pada *single destination* yang akhirnya berpengaruh pada nilai *delivery probability*nya.

5.2 Overhead Ratio

Overhead Rotio merupakan berapa banyak paket redudansi terkait untuk peiriman satu pesan ke tujuan. Apabila nilai dari *Overhead Rotio* semakin rendah maka semakin optimal protokol *routing* yang digunakan. Berikut adalah tabel nilai *Overhead Ratio* hasil pengujian dari skenario ditunjukan pada tabel 5.3 dan 5.4.

Tabel 5.3 Hasil nilai *Overhead Ratio* pada skenario *Single Destination*

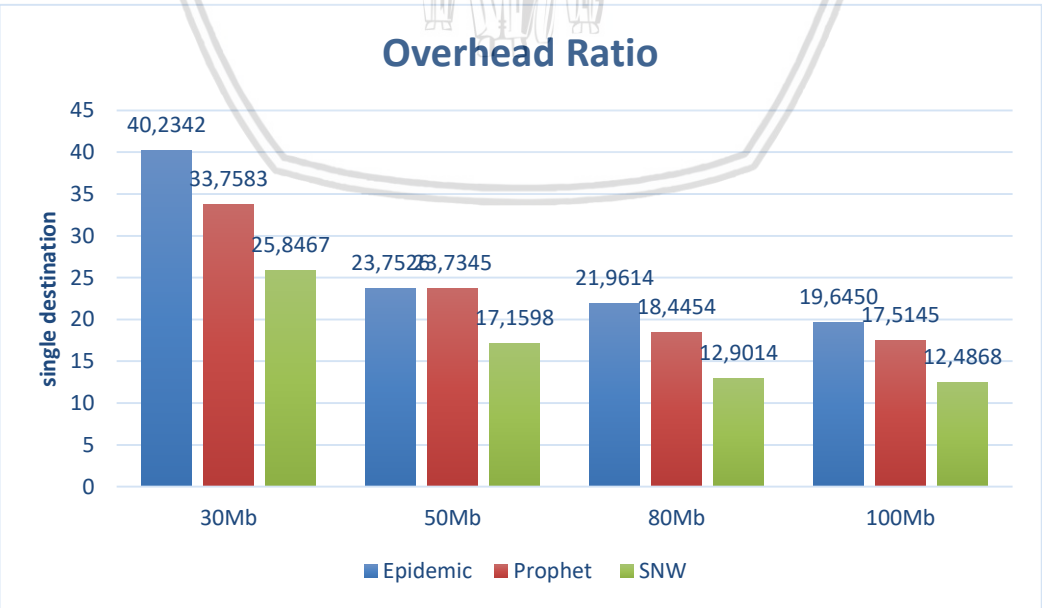
Overhead Ratio			
Buffer	Epidemic (%)	Prophet (%)	SNW (%)
30Mb	40,2342	33,7583	25,8467
50Mb	23,7526	23,7345	17,1598
80Mb	21,9614	18,4454	12,9014
100Mb	19,6450	17,5145	12,4868

Pada tabel 5.3 menunjukan hasil nilai *Overhead Ratio* dari pengujian skenario 1, 3 dan 5 dengan mekanisme skenario *Single Destination.* Semua skenario pengujian dijalankan selama 12 jam waktu simulator dengan *node* bergerak sebanyak 150 dengan kecepatan 10 – 50 Km/jam, *Stationary Relay Node* sebanyak 10 serta *Management Buffer* FIFO dengan ukuran pesan sebesar 5Mb dan ukuran *buffer* sebesar 30Mb, 50Mb, 80Mb dan 100Mb.

Tabel 5.4 Hasil nilai *Overhead Ratio* pada skenario *Multi Destination*

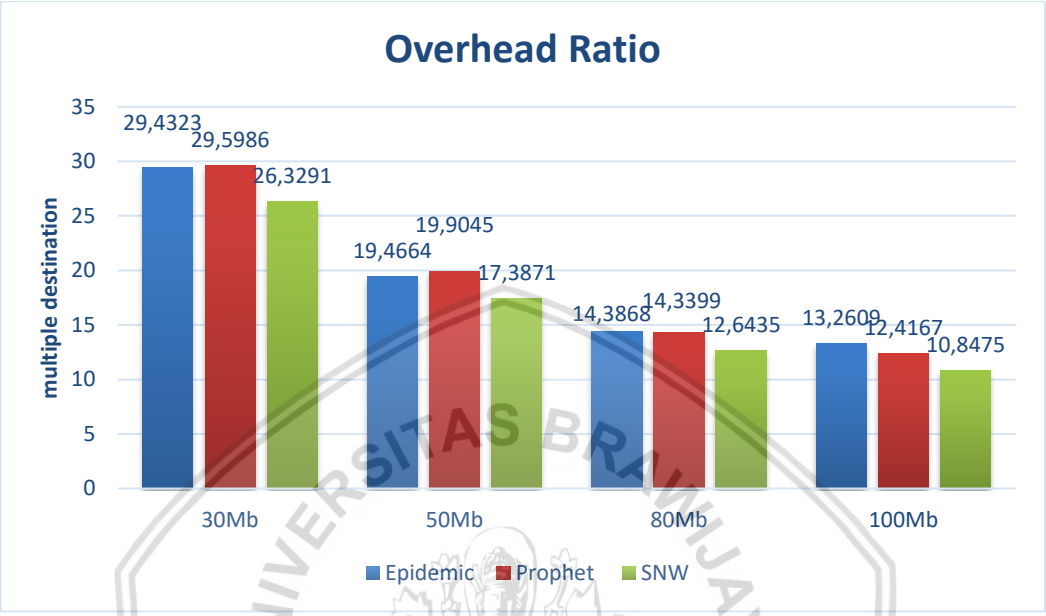
Overhead Ratio			
Buffer	Epidemic (%)	Prophet (%)	SNW (%)
30Mb	29,4323	29,5986	26,3291
50Mb	19,4664	19,9045	17,3871
80Mb	14,3868	14,3399	12,6435
100Mb	13,2609	12,4167	10,8475

Pada tabel 5.4 menunjukan hasil nilai *Overhead Ratio* dari pengujian skenario 2, 4 dan 6 dengan mekanisme skenario *Multiple Destination.* Semua skenario pengujian dijalankan selama 12 jam waktu simulator dengan *node* bergerak sebanyak 150 dengan kecepatan 10 – 50 Km/jam, *Stationary Relay Node* sebanyak 10 serta *Management Buffer* FIFO dengan ukuran pesan sebesar 5Mb dan ukuran *buffer* sebesar 30Mb, 50Mb, 80Mb dan 100Mb.



Gambar 5.3 Grafik *Overhead Ratio* pada skenario *Single Destination*

Pada grafik 5.3 diatas merupakan bentuk data diagram yang dibuat berdasarkan tabel 5.3 dari pengujian skenario menggunakan ukuran *buffer* 30Mb, 50Mb, 80Mb dan 100Mb dengan protokol *routing Epidemic, ProPHET* dan *Spray And Wait* terhadap parameter uji *Overhead Ratio* yang didapat berdasarkan skenario pengiriman pesan dengan mekanisme *Single Destination.*



Gambar 5.4 Grafik *Overhead Ratio* pada skenario *Multi Destination*

Pada grafik 5.4 diatas merupakan bentuk data diagram yang dibuat berdasarkan tabel 5.4 dari pengujian skenario menggunakan ukuran *buffer* 30Mb, 50Mb, 80Mb dan 100Mb dengan protokol *routing Epidemic, ProPHET* dan *Spray And Wait* terhadap parameter uji *Overhead Ratio* yang didapat berdasarkan skenario pengiriman pesan dengan mekanisme *Multiple Destination.*

Berikut ini merupakan penjelasan dan analisis dari grafik 5.3 dan 5.4 berdasarkan protokol *routing Epidemic, ProPHET* dan *Spray And Wait* pada skenario pengiriman pesan dengan mekanisme *single destination* dan *multiple destianion*.

1. *Routing Epidemic*

   Pada grafik gambar 5.3 dan gambar 5.4 menunjukan *routing Epidemic* terjadi penurunan *overhead ratio* pada setiap ukuran *buffer* pada skenario pengiriman *single destination* maupun *multiple destianion.* pada ukuran *buffer* 30Mb *routing epidemic* memiliki *overhead ratio* sebesar 40,2% dan 29,4%. Kemudian pada ukuran *buffer* 50Mb *routing epidemic* terjadi penurunan dengan memiliki nilai *overhead ratio* sebesar 23,8% dan 19,5%. Selanjutnya pada ukuran *buffer* 80Mb *routing epidemic* terjadi penurunan dengan memiliki nilai *overhead ratio* sebesar 21,9 dan 14,4%. Yang terakhir pada ukuran *buffer* 100Mb *routing epidemic* terjadi penurunan dengan memiliki nilai *overhead ratio* sebesar 19,6% dan 13,2%.

Pada kenyataannya *routing Epidemic* memiliki nilai *Overhead Ratio* yang cukup tinggi, hal ini dikarenakan beban jaringan dari mode penyebaran pesan pada *routing Epidemic.* Pesan akan terus disebarkan dari *node* satu ke *node* lainnya sehingga pada ukuran *buffer* yang terbatas *routing Epidemic* memiliki *Overhead Ratio* yang tinggi. Untuk itu penggunakan *management buffer* dapat mengurangi beban jaringan pada *Epidemic*. Ukuran *buffer* yang terbatas dapat dimaksimalkan dengan penerapan *management buffer*. Pada penelitian ini algoritma FIFO digunakan untuk me-*management buffer* sehingga saat *buffer* penuh maka pesan yang tersimpan pada *node buffer* dapat dihapus untuk mengakomodasi pesan baru, dengan adanya *management buffer* maka nilai Overhead *Ratio* pada *routing Epidemic* dapat berkurang.

Selain *management buffer* yang digunakan dapat mengurai nilai *overhead ratio* pada *routing epidemic*, ukuran *buffer* yang lebih besar juga dapat mengurai kepadatan jaringan pada *epidemic*. hal ini karena penyimpanan *node buffer* dapat menampung lebih banyak pesan untuk dikirimkan ke tujuan. R*outing Epidemic* pada ukuran *buffer* 30Mb, 50Mb, 80Mb dan 100Mb memiliki penurunan nilai *Overhead Ratio,* sehingga bisa disimpulkan semakin besar ukuran *buffer* yang digunakan maka semakin rendah nilai *overhead ratio* yang ditunjukan oleh *routing Epidemic.* Hal ini menunjukan beban yang terjadi pada jaringan juga semakin kecil.

Dapat diamati perbedaan nilai *Overhead Ratio* pada skenario *Single Destination* dan *Multiple Destination,* bahwa pada *Single Destination* nilai *Delivery Probability*nya lebih tinggi dari pada *Multiple Destination* yang terjadi pada semua ukuran *buffer*. Hal ini terjadi karena pada skenario *Multiple Destination* pesan yang dibuat untuk dikirimkan sejumalah 2468 jumlahnya dua kali lipat lebih banyak dari pada pesan yang dibuat pada skenario *Single Destination* yang hanya 1234 tetapi dengan batasan *buffer* yang sama. Sehingga *multiple destination* pada *routing Epidemic* lebih efektif dalam mengurangi nilai *Overhead Ratio* tetapi tidak efektif dalam menyampaikan pesan. Hal ini terjadi karena perhitungan *Overhead Ratio* berdasarkan jumlah pesan yang terkirim dan jumlah pesan yang disalinkan, sehingga banyaknya pesan yang dibuat tidak berpengaruh pada nilai *Overhead Ratio*nya. Pada ukuran *buffer* 30Mb, 50Mb, 80Mb dan 100Mb skenario *multiple destination* menyampaikan pesan ke tujuan sebanyak 141, 238, 318 dan 345 dengan jumlah pesan yang tersalin sebanyak 4719, 4871, 4954 dan 4920. Sedangkan pada skenario *single destination* dengan ukuran *buffer* yang sama yaitu 30Mb, 50Mb, 80Mb dan 100Mb menyampaikan pesan sebanyak 111, 190, 207 dan 231 dengan jumlah pesan yang tersalin sebanyak 4577, 4703,4703 dan 4769. Hal ini yang menjadikan *multiple destination* memiliki nialai *overhead ratio* lebih kecil dari pada *single destination*.

2. *Routing ProPHET*

Pada grafik gambar 5.3 dan gambar 5.4 menunjukan *routing ProPHET* terjadi penurunan *overhead ratio* pada setiap ukuran *buffer* pada skenario pengiriman *single destination* maupun *multiple destianion.* pada ukuran

buffer 30Mb *routing ProPHET* memiliki nilai *overhead ratio* sebesar 33,76% dan 29,6%. Kemudian pada ukuran *buffer* 50Mb *routing ProPHET* terjadi penurunan dengan memiliki nilai *overhead ratio* sebesar 23,7% dan 19,9%. Selanjutnya pada ukuran *buffer* 80Mb *routing ProPHET* terjadi penurunan dengan memiliki nilai *overhead ratio* sebesar 18,4% dan 14,3%. Yang terakhir pada ukuran *buffer* 100Mb *routing ProPHET* terjadi penurunan dengan memiliki nilai *overhead ratio* sebesar 17,5% dan 12,4%.

Pada skenario pengujian *Single Destination* dengan *buffer* sebesar 50 Mb serta pada skenario pengujian *Multi Destination* dengan *buffer* 30Mb dan 50Mb *routing ProPHET* memiliki nilai *Overhead Ratio* tertinggi hal ini terjadi karena besarnya beban pada jaringan. Tingginya peningkatan nilai *Overhead Ratio* pada *ProPHET* juga dapat terjadi karena perhitungan *delivery predictability* yang berlebihan. Hal ini akan menyebabkan banyaknya *node* yang akan memiliki nilai *delivery predictability* yang tinggi, sehingga penyebaran pesan terjadi kebanyak *node* yang sebenarnya *node* tersebut tidak optimal untuk meneruskan pesan sampai ke tujuan sehingga beban jaringan akan meningkat pada ukuran *buffer* yang terbatas. Untuk itu penggunaan *management buffer* sangat membantu dalam menekan tingginya *Overhead Ratio* pada *routing ProPHET*.

Management buffer dapat berperan untuk mengelola pesan pada penyimpanan *buffer node* dengan keterbatasan ukuran *buffer*, sehingga pesan yang disalinkan ke *node* tertentu dapat terakomodasi meskipun penyimpanan *buffer* telah penuh. Pada skenario pengujian dengan ukuran *buffer* 30Mb, 50Mb, 80Mb dan 100Mb *routing ProPHET* memiliki penurunan nilai *Overhead Ratio* sehingga bisa disimpulkan bahwa semakin besar ukuran *buffer* yang digunakan maka dapat menurunkan nilai *Overhead Ratio*, hal ini karena penyimpanan pada *buffer* yang besar dapat menyimpan dan meneruskan pesan lebih banyak sehingga beban jaringan dapat berkurang.

Dapat diamati perbedaan nilai *Overhead Ratio* pada skenario *Single Destination* dan *Multiple Destination,* bahwa pada *Single Destination* nilai *Delivery Probability*nya lebih tinggi dari pada *Multiple Destination* yang terjadi pada semua ukuran *buffer*. Hal ini terjadi karena pada skenario *Multiple Destination* pesan yang dibuat untuk dikirimkan sejumalah 2468 jumlahnya dua kali lipat lebih banyak dari pada pesan yang dibuat pada skenario *Single Destination* yang hanya 1234 tetapi dengan batasan *buffer* yang sama. Pada penyebaran pesannya *routing ProPHET* memprioritaskan salinan pesan ke *node* yang mempunyai tingkat *delivery predictability* tinggi, sehingga *multiple destination* pada *routing ProPHET* lebih efektif dalam mengurangi nilai *Overhead Ratio* tetapi tidak efektif dalam menyampaikan pesan. Hal ini terjadi karena perhitungan *Overhead Ratio* berdasarkan jumlah pesan yang terkirim dan jumlah pesan yang disalinkan, maka banyaknya pesan yang dibuat tidak berpengaruh pada nilai *Overhead Ratio*nya. Pada ukuran *buffer* 30Mb, 50Mb, 80Mb dan 100Mb skenario *multiple destination* menyampaikan pesan ke tujuan sebanyak 158, 220, 303 dan 348 dengan jumlah pesan yang tersalin

sebanyak 4345, 4599, 4648 dan 4669. Sedangkan pada skenario *single destination* dengan ukuran *buffer* yang sama yaitu 30Mb, 50Mb, 80Mb dan 100Mb menyampaikan pesan sebanyak 120, 177, 229 dan 241 dengan jumlah pesan yang tersalin sebanyak 4171, 4378, 4453 dan 4462. Hal ini yang menjadikan *multiple destination* memiliki nialai *overhead ratio* lebih kecil dari pada *single destination*.

3. *Routing Spray And Wait*

Pada grafik gambar 5.3 dan gambar 5.4 menunjukan *routing Spray And Wait* terjadi penurunan *overhead ratio* pada setiap ukuran *buffer* pada skenario pengiriman *single destination* maupun *multiple destianion.* pada ukuran *buffer* 30Mb *routing Spray And Wait* memiliki nilai *Overhead Ratio* sebesar 25,8% dan 26,3%. Kemudian pada ukuran *buffer* 50Mb *routing Spray And Wait* terjadi penurunan dengan memiliki nilai *Overhead Ratio* sebesar 17,1% dan 17,4%. Selanjutnya pada ukuran *buffer* 80Mb *routing Spray and Wait* terjadi penurunan dengan memiliki nilai *overhead ratio* sebesar 12,9% dan 12,6%. Yang terakhir pada ukuran *buffer* 100Mb *routing Spray and Wait* terjadi penurunan dengan memiliki nilai *overhead ratio* sebesar 12,5% dan 10,8%.

Dari percobaan diatas diketahui nilai *Overhead Ratio* pada *routing Spray And Wait* memiliki nilai terkecil dari *routing Epidemic* dan *ProPHET,* baik dalam skenario pengujian *single Destination* maupun *multiple Destination.* Hal ini terjadi karena *routing Spray and Wait* memiliki dua tahap dalam pengiriman. Tahap pertama ialah tahap *Spray.* Pada tahap ini *routing Spray and Wait* melakukan penyebaran pesan ke semua *node* seperti pada *routing Epidemic* tetapi memiliki batasan penyebaran dengan jumlah tertentu, pada pengujian ini batas penyebaran sebanyak 6 *copy*. Selanjutnya apabila pada tahap *spray node* tujuan tidak ditemukan dan jumlah penyebaran telah mencapai batasan, maka tahap ke dua yaitu tahap *wait* akan berjalan yakni pesan akan dibawa sampai bertemu dengan *node* tujuan. Dengan demikian penyebaran pesan pada jaringan dapat dibatasi agar beban pada jaringan tidak meningkat tetapi nilai *Delivery Probability* tidak berkurang secara signifikan karena pada tahap *spray* membuat banyak *node* yang memiliki salinan pesan untuk dikirimkan ke tujuan. Selain itu *management buffer* dapat membantu dalam mengurangi nilai *Overhead Ratio* pada *routing Spray and Wait*.

Management buffer dapat berperan dalam fase *Spray* saat penyebaran pesan yang secara terus menerus dilakukan. *Management buffer* berperan untuk mengelola pesan pada penyimpanan *buffer node* dengan keterbatasan ukuran *buffer*, sehingga pesan yang disalinkan ke *node* tertentu dapat terakomodasi meskipun penyimpanan *buffer* telah penuh. Selain itu pada *buffer* 30Mb, 50Mb, 80Mb dan 100Mb *routing Spray and Wait* memiliki penurunan nilai *Overhead Ratio* dalam skenario pengujian *single Destination* maupun *multiple Destination* sehingga bisa disimpulkan bahwa semakin besar ukuran *buffer* yang digunakan maka dapat menurunkan nilai *Overhead Ratio* pada *routing Spray and Wait* karena semakin besar ukuran *buffer* yang

digunakan maka penyimpanan pesan pada *buffer node* saat pendistribusian pesan akan semakin banyak juga.

Pada *routing Spray and Wait* nilai *Overhead Ratio* skenario *Single Destination* dengan ukuran *buffer* 30Mb dan 50Mb memiliki nilai *Overhead Ratio* lebih rendah daripada skenario *Multiple Destination*. Akan tetapi, pada *buffer* lebih tinggi yakni 80Mb dan 100Mb skenario *Multiple Destination* memiliki nilai *Overhead Ratio* yang lebih rendah. Hal ini terjadi karena *routing Spray and Wait* memiliki dua fase dalam pengiriman yaitu fase pertama ialah tahap *Spray* yang mana pada tahap ini *routing Spray and Wait* melakukan penyebaran pesan ke semua *node.* Fase ke dua yaitu fase *wait* dengan mekanisme membawa pesan sampai bertemu dengan *node* tujuan. Pada fase wait besarnya ukuran *buffer* sangat memengaruhi jumlah pesan yang tersampaikan ke tujuan, apabila semakin besar ukuran *buffer* maka pesan yang sampai ketujuan akan semakin banyak dan akan mempengaruhi nilai *overhead ratio*. Selain itu, perhitungan *Overhead Ratio* berdasarkan jumlah pesan yang terkirim dan jumlah pesan yang disalinkan, sehingga banyaknya pesan yang dibuat tidak berpengaruh pada nilai *Overhead Ratio*nya.

Pada ukuran *buffer* 30Mb, 50Mb, 80Mb dan 100Mb skenario *multiple destination* menyampaikan pesan ke tujuan sebanyak 158, 248, 345 dan 400 dengan jumlah pesan yang tersalin sebanyak 4318, 4560, 4707 dan 4739. Sedangkan pada skenario *single destination* dengan ukuran *buffer* yang sama yaitu 30Mb, 50Mb, 80Mb dan 100Mb menyampaikan pesan sebanyak 137, 219, 294 dan 304 dengan jumlah pesan yang tersalin sebanyak 3678, 3977, 4087 dan 4100. Hal ini yang menjadikan *multiple destination* memiliki nilai *overhead ratio* lebih kecil dari pada *single destination* pada ukuran *buffer* 30Mb dan 50Mb tetapi lebih besar pada ukuran *buffer* 70Mb dan semakin besar pada *buffer* 100Mb. Sehingga dapat disimpulkan bahwa skenario *multiple destination* lebih efektif untuk mengurangi nilai *Overhead Ratio* seiring bertambah besarnya ukuran *buffer* yang digunakan.

BAB 6 PENUTUP

6.1 Kesimpulan

Berikut adalah kesimpulan yang diambil untuk menjawab masalah yang diangkat pada penelitian ini.

1. Untuk merancang dan mensimulasikan teknologi DTN pada *ONE Simulator* yaitu dengan pembuatan jalur peta untuk jalur pergerakan *node* yang diambil dari *Open Street Map* yang kemudian dijadikan file dengan format .wkt, penentuan koordinat *Stationary Relay Node* untuk menentukan letak *Stationary Relay Node* sepanjang jalur pengiriman pesan. Kemudian konfigurasi *ONE Simulator* yang meliputi instalasi *ONE Simulator,* melakukan konfigurasi waktu skenario dengan waktu simulasi selama 12 jam, menentukan pergerakan *node* sepanjang jalur pengiriman, menentuan protokol *routing* yang akan digunakan yaitu *Epidemic, ProPHET* dan *Spray and Wait,* menentukan jumlah *node* sebagai media pengiriman yang bergerak yaitu sebanyak 150 *node* dengan kecepatan bergerak 10 – 50 Km/jam, kemudian memberikan batasan ukuran *buffer,* menentukan *Time To Live* dari pesan yang didistribusikan yaitu selama 420 detik, melakukan konfigurasi *source and destination* pesan kemudian melakukan konfigurasi *management buffer* dengan algoritma *first in – first out.*

2. Hasil nilai *Delivery Probability* pada protokol *routing Spray and Wait* yang dihasilkan memiliki nilai teringgi dari pada protokol *routing Epidemic* dan *ProPHET* baik dalam skenario pengujian *Single Destination* dan *Multiple Destination* pada setiap ukuran *buffer* yang diujikan. Pada *buffer* 30Mb, 50Mb, 80Mb dan 100Mb protokol *routing Spray and Wait* memiliki nilai *Delivery Probability* terbesar yaitu sebesar 0,1110, 0,1775, 0,2382 dan 0,2464 pada skenario *Single Destination,* sedangkan pada skenario *Multiple Destination* memiliki nilai sebesar 0,0640, 0,1005, 0,1398 dan 0,1621. Hal ini terjadi karena *routing Spray and Wait* memiliki dua tahap dalam pengiriman pesan yaitu tahap *spray* dan tahap *wait*. Pemakaian *Stationary Relay Node* juga dapat menaikan nilai *Delivery Probability* karena memberikan jaminan untuk setiap *node* dapat saling bertemu sehingga pada tahap *Spray* salinan pesan ke *node* lain akan semakin banyak.

   Pada protokol *routing Spray and Wait* nilai *Overhead Ratio* yang dihasilkan memiliki nilai terendah pada skenario pengujian *Single Destination* dan *Multiple Destinatio*. Pada *buffer* 30Mb, 50Mb, 80Mb dan 100Mb protokol *routing Spray and Wait* terjadi penurunan nilai *Overhead Ratio* dan masing – masing memiliki nilai *Overhead Ratio* sebesar 25,85%, 17,16%, 12,94% dan 12,49% pada skenario *Single Destination,* sedangkan pada skenario *Multiple Destination* memiliki nilai sebesar 26,33%, 17,39, 12,6% dan 10,85%. Pada setiap pengujian dengan *buffer* yang berbeda – beda selalu mengalami penurunan nilai *overhead ratio,* hal ini dikarenakan *management buffer* FIFO

yang digunakan pada penelitian ini dapat mengurangi beban pada jaringan, sehingga nilai overhead ratio dapat menurun.

Dari poin – poin diatas dapat disimpulkan bawah dalam penelitian ini *routing Spray and Wait* lebih efektif dalam penyampaian pesan dengan nilai *Delivery Probabiliti* yang tinggi dan *Overhead Ratio* yang rendah dari pada *routing Epidemic* dan *ProPHET* meskipun dengan kondisi *buffer* yang terbatas.

6.2 Saran

Untuk penelitan *Delay Tolerant Network* yang selanjutnya disarankan untuk menambahkan protokol *routing* DTN dengan mekanisme multi jalur dan menambahkan *management buffer* yang berbeda atau dengan membandingkan beberapa *management buffer* dan pergerakan node yang berbeda – beda.

DAFTAR PUSTAKA

Bhattacarya, S. S., Derperette, E. F. & Teich, J., 2003. *Domain-Specific Processors.* India: CRC Press.

Fajri, M., 2016. Simulasi Antrian Paket Data Jaringan. *Jurnal Ilmiah FIFO,* Volume Volume VIII/No. 2/November/2016, p. 151.

Filhoa, J. G., Patel, A. & Batist, B. L. A., 2016. A systematic technical survey of DTN and VDTN routing protocols. *Computer Standards & Interfaces,* Volume 48, pp. 139-159.

Gamit & Patel, H., 2014. Evaluation Of DTN Routing Protocol. *International Journal Of Engineering Sciences and Research Technology,* 3(2), pp. 588-592.

IACC, 2016. *International Conference On Advances In Computing & Communications.* India: Cochin.

LAP, 2010. *Delay Tolerant Network.* Germany: LAP Lambert Academic Publishing.

Leanna VY & NR, J., 2016. Analisis Performansi Algoritma Routing First Contact dengan Stationary Relay Node pada Delay Toleran Network. *Jurnal Elkonomika,* 4(2), pp. 123-133.

Mehto, A., 2013. *Comparing Delay Tolerant Network Routing Protocols for Optimizing L-Copies in Spray and WaitRouting for Minimum Delay.* Malaysia: Conferense on Advances in Communication and Control System.

Muis, A., 2013. *Optimization Performance Buffer Management in Delay Tollerant Network Using Network Routing Protocols Multicopy.* Makassar: Universitas Indonesia Timur Makassr.

Poltak, H. G., 2017. Perbandingan Kinerja Routing Multi Copy dan Routing First Contact dengan Stationary Relay Node pada Delay Tolerant Network (DTN). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* 2(7), pp. 2513-2522.

Restu, J. N. & Yovita, L. V., 2015. *Analisis Performansi Vehicular Ad Hoc Network Menggunakan Protokol Routing DTN Direct Delivery dan First Contact..* Yogyakarta: Seminar Nasional Universitas PGRI Yogyakarta..

Spyropoulos, T., Psounis, K. & Raghavendra, C., 2004. Single-copy routing in intermittently connected mobile networks. *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks,,* pp. 235-244.

Wahanani, H. S. I. A. D., 2015. *Analisa kinerja protokol routing Delay Tolerant Network (DTN) untuk transportasi publik.* Surabaya: UPN Vetaran Jawa Timur..

Warthman, F., 2012. *Delay- and Disruption-Tolerant Networks (DTNs).* United States of America: Warthman Associates.

Widhiyanto, A. S., 2016. *Analisis untuk Kerja Protokol Routing Rapid di Jaringan Oportunistik [Skripsi].* Yogyakarta: Universitas Sanatha Darma.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA